# Oracle® Utilities Market Settlements Management Cloud Service

## User Guides

Release 26.10
G58682-01
October 2026

Oracle Utilities Market Settlements Management Cloud Service User Guides, Release 26.10

G58682-01

Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

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

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

# Contents

Part I Oracle Utilities Market Settlements Management Cloud Service User Guides

Part II Meter Solution Business User Guide

Part III Application Framework Administrative User Guide

Part IV Meter Solution Administrative User Guide

Part V Oracle Utilities Cloud Service Foundation


Copyright © 2026, Oracle and/or its affiliates.

# Part I

# Oracle Utilities Market Settlements Management Cloud Service User Guides

Oracle Utilities Market Settlements Management Cloud Service shares significant functionality with Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway. As such, they all use the same user documentation. Oracle Utilities Market Settlements Management Cloud Service User Guides comprise the following:

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

Refer to Settlement in the *Business User Guide* and Defining Settlement Options in the *Administrative User Guide* for specific information related to Oracle Utilities Market Settlements Management Cloud Service.


Copyright © 2026, Oracle and/or its affiliates.

# Part II

# Meter Solution Business User Guide

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

- Oracle Utilities Meter Data Management
- Oracle Utilities Smart Grid Gateway
- Oracle Utilities Market Settlements Management
- Oracle Utilities Meter Solution Cloud Service
- Oracle Utilities Market Settlements Management Cloud Service

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


- Overview
- User Interface Features
- Asset Management
- Device Management
- Device Installations
- Measurements
- VEE
- 360 Degree and Unified Search
- 360 Degree Views
- Usage
- Communications
- Settlement
- Integrations
- Aggregations
- Master Data Sync
- Outage Storm Mode
- Reports
- Dashboards
- Totals and Trends
- Time Zone Support
- Service Issue Monitors
- Service Order Management
- Data Access Restrictions
- To Do Processing




Copyright © 2026, Oracle and/or its affiliates.

- Reports


# Overview


- Functional Overview
- Glossary of Terms
- High Level Administration Areas
- Additional Resources


## Functional Overview

Oracle Utilities meter solution products provides functionality for handling large volumes of meter/device data to enable increased accuracy, flexibility, and scalability. Oracle Utilities meter solution products include:

- Oracle Utilities Meter Data Management
- Oracle Utilities Smart Grid Gateway
- Oracle Utilities Service Order Management
- Oracle Utilities Market Settlements Management

Below is a list of the most crucial business processes that are enabled within the system:

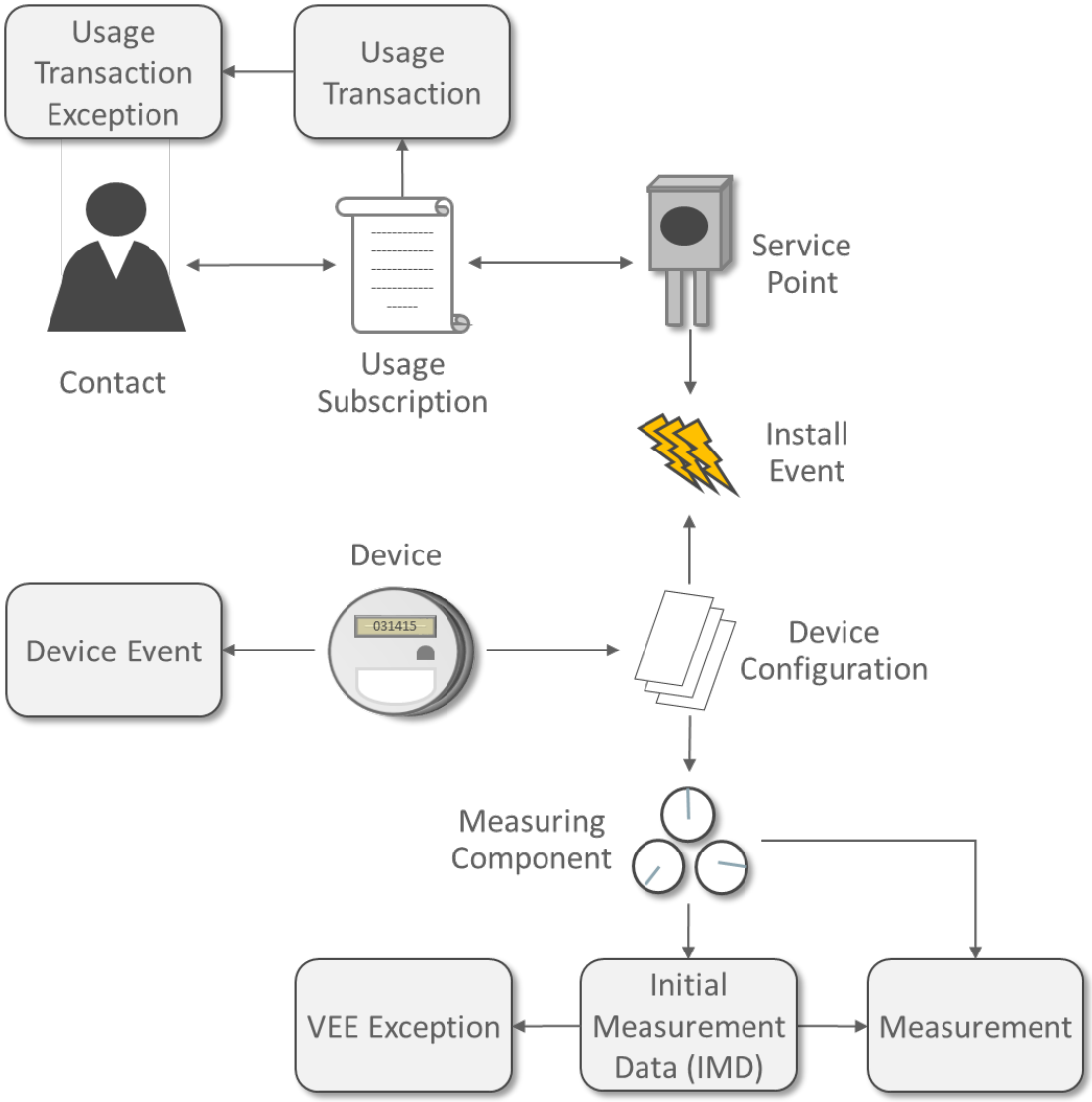
- Defining meters, meter configurations, service points, and meter installations
- Loading of meter readings and interval data from a head-end system or other source
- Automatic validation, editing, and estimation of measurement data
- Robust editing capabilities for readings and interval data
- Calculation and publishing of bill determinants and other data from measurement data for use in external down-stream systems such as billing, pricing, etc.

Oracle Utilities meter solution products store a lot of important data that varies from meters to service points to customer contacts and everything in between. The diagrams below provides an overview of the most crucial objects in the Oracle Utilities meter solution data model:

**Measurement, VEE, and Usage Calculation for Billing**



Copyright © 2026, Oracle and/or its affiliates.




A Device represents a physical meter, communication module, or some other device out in the field. A Service Point is the point at which service is delivered to a customer and a device can be installed. The Install Event is a record of a specific device installed at a Service Point. A Contact is the customer that should be contacted in association with the Usage Subscription.

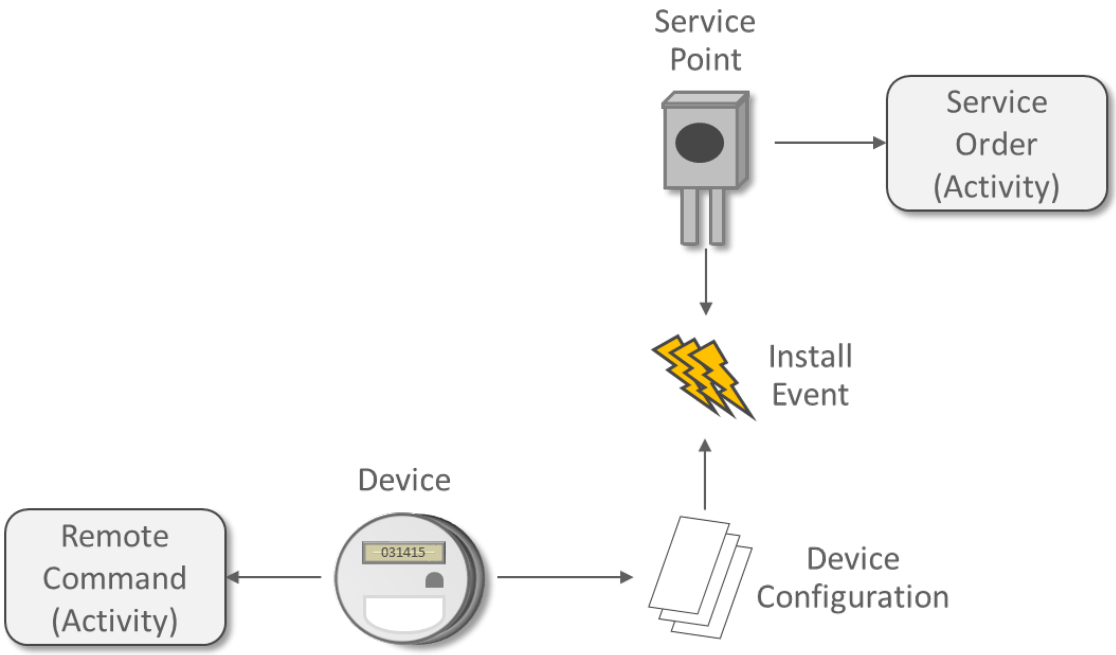
In order to properly track the way in which a device is configured, the Device Configuration (add link) keeps a record of which types of data should be measured for the device. A Measuring Component (add link) represents a single channel of data for a device. For example, a device may have multiple measuring components: one that represents kWh interval data and another, another that represents kWh scalar readings, and a third that registers Voltage interval data.

Energy data received from meters is initially stored as Initial Measurement Data (IMD). Once the VEE process is executed, if the data passes then Measurements are created but if the data fails then VEE Exceptions are created.

A Usage Subscription tracks a set of usage calculations that should be performed for a Service Point or multiple Service Points. In order to perform calculation of usage (often referred to as bill determinants), a Usage Transaction is created through a request from the billing system. If any issues are encountered in the usage calculation process or from a usage transaction validation, then Usage Transaction Exceptions are created.


Copyright © 2026, Oracle and/or its affiliates.

**Service Orders and Remote Commands**




An Activity in Oracle Utilities Meter Data Management is a very flexible object that is used for various processes. Two of the key processes handled by Activities are:

- Remote Commands involve communicating with a head end system to perform actions or retrieve data from a meter. Examples include remote connects, remote disconnects, on-demand reading, etc.
- Service Orders are methods through which work can be performed at a Service Point. Examples include enabling service, disabling service, meter exchanges, etc.

Oracle Utilities meter solution products include the following functional areas:

- **Asset Management** is used by analysts and administrators to manage and define the assets used to record and capture meter data. Optional.
- **Device Management** is used by analysts and administrators in managing and defining the devices used to record and capture meter data.
- **Device Installation** is used by analysts and administrators in managing the installation of devices, including defining markets and service providers, Service Points and contacts, and installation events.
- **Measurement data** is a normalized way of storing data from a meter that involves some form of measurement (i.e. kWh, CCF, etc.). Both interval data and scalar readings are held in this common storage location.
- **Validation, Editing, and Estimation (VEE)** is used by administrators to define validation, editing, and estimation (VEE) rules to be applied to measurement data. VEE Exceptions may result from validation or estimation failures and should be worked by analysts through the To Do process.


Copyright © 2026, Oracle and/or its affiliates.

- **360 Degree Search and View** is used by analysts and administrators to search and view data for devices, measuring components, Service Points, Usage Subscriptions, and contacts.
- **Consumption Sync** provides an automatic method to keep interval data and scalar readings in sync with one another.
- **Usage Calculation** is used by administrators to manage the calculation of usage data and to provide the results of those calculations (commonly referred to as bill determinants) to external systems and parties. Usage calculation groups and rules define calculation rules to be applied to measurement data. Usage Transaction Exceptions may result from usage calculation and should be worked by analysts through the To Do process.
- **Device Events** provides a view of specific events that have occurred on a meter. These are often unexpected and can indicate an issue with the meter.
- **Communication** helps track the instances when communication occurs with external systems. This is heavily used for tracking remote commands against Head End systems.
- **Aggregations** are used by analysts and administrators to search, view and maintain aggregated measurements that represent a summarization of other measurements from a set of devices and/or measuring components.
- **Master Data Sync** defines the methods through which data is automatically synchronized to Oracle Utilities Meter Data Management from external sources (such as a CIS and/or Asset Management system).
- **Outage Storm Mode** is a way for Oracle Utilities Meter Data Management to detect widespread outages and suppress estimation for those meters until normal communication of data resumes.
- **Service Order Management** provides a centralized program for managing the complex interactions required for Service Order processing. This area is especially valuable for managing service order processing that involves remote communication with a Head End for connects, disconnects, and on-demand readings.
- **Market Settlement** provides the core functionality for calculating and settling energy changes.
- **Service Issue Monitor** can be setup by administrators to monitor various conditions within Oracle Utilities Meter Data Management and automatically create a Service Investigative Order if those conditions are met.
- **Dashboards** provide high level metrics for Oracle Utilities Meter Data Management operators to monitor operational trends as well as overall system health.
- **Information Lifecycle Management** is an automated method that administrators can configure to prepare data for archiving or purging after a defined period of time for the record type.

# Glossary of Terms

Below is a list of terms along with their definition to assist with understanding Oracle Utilities Meter Data Management:

| Term | Definition |
|---|---|
| Base Time Zone | This is the time zone defined on the Installation Options - Framework. All date/times stored in the system are governed by this time zone. See also: Standard Time and Legal Time |


Copyright © 2026, Oracle and/or its affiliates.

| Term | Definition |
| --- | --- |
| Bill Determinants | Measurement data summarized for use by a billing application. Bill determinants can take the form of TOU-mapped interval consumption, scalar consumption, scalar readings, and/or interval consumption obtained via measurements. One good example of bill determinant is TOU-mapped interval consumption, which reduces a full month's worth of interval data into several buckets of consumption based on time of use. |
| CIS (Customer Information System) | A generic term for the system used to store customer and billing information at a utility. |
| Consumption | A measurement by a given device of the amount of energy, water, gas, etc. consumed over a given time period. Synonymous with the term "measurement". |
| Contact | A customer that should be contacted in association with a Service Point or Usage Subscription. |
| Customer Modification (CM) | This refers to code or configuration that's been added by your organization. |
| Device | A physical meter, communication module, or some other device out in the field. |
| Device Configuration | This keeps a record of which types of data should be measured for the device. |
| Device Event | An alarm or warning sent from a device that may indicate an issue with normal operations (examples: theft, memory failures, etc.). |
| Dynamic Option | Used to specify terms that override how usage is normally calculated - such as a critical peak period that affects the TOU mapping of interval consumption. |
| Factor | A centrally stored set of values for use in validation rules, bill determinants calculations, and other processes. A factor can have different values depending upon some definable attribute of a system object, such as customer size associated with a Service Point. The values are effective-dated so that changes over time are retained. Examples of factors can include minimum/maximum thresholds, loss factors, etc. Classes of factors are defined that can have numeric values (as in the above examples), or values pointing to profile measuring components or VEE groups. |
| Function - Measurement | An online-initiated action applied to measurement data, comprising one or more measurement services. |
| Initial Measurement Data (IMD) | The data measured on a device that is sent to Oracle Utilities Meter Data Management is stored in this initial record while it's being processed. |


Copyright © 2026, Oracle and/or its affiliates.

| Term | Definition |
| --- | --- |
| Install Event | An instance of a specific device installed at a Service Point. This also includes a record of any time the service was turned on or off. |
| Legal Time | This refers to a method of storing date/time information where there is an adjustment made to the date/time during periods where Daylight Savings Time (DST) is in effect. There are two offsets against Universal Coordinated Time (UTC): (1) when DST is in effect, (2) when DST is not in effect.<br>On the day that day that DST begins the clock moves forward by a period of time equal to the DST offset. For example, if DST takes effect at 2:00AM and the offset is one hour then the clock moves from 1:59:59AM to 3:00:00AM skipping 2:00:00AM - 2:59:59AM.<br>On the day that DST ends the clock is moved backwards thus recording a period of time equal to the DST offset twice, the first recording of that period is with the UTC offset for DST being in effect, the second recording of that period is with the UTC offset for DST not being in effect. For example, if DST ends at 2:00AM and the offset is one hour then the clock would repeat the hour from 1:00:00 to 1:59:59. |
| Manufacturer | The company that manufactured a device. |
| Measurement | This is the final, validated measurement information from a device. Measurement data is used in calculation of usage as part of the Usage Transaction process. |
| Measuring Component | A logical container for measurement data. These are often channels for physical devices but can also be aggregators, weather data, or any other form of measurements. |
| Out-of-the-box (OOTB) | This refers to code or configuration that's provided as part of the product delivered by Oracle (as opposed to a Customer Modification). |
| Profiling of Scalar Data | The process of applying an interval consumption "shape" to a scalar measurement, using an existing interval measuring component. The individual interval values are adjusted such that when totaled, they equal the value of the scalar measurement. |
| Service Point | A location at which a company supplies service. Used to store information describing the type of service and how it is measured. |
| Service Quantity Identifier (SQI) | Used to further distinguish between measured quantities that have identical UOM/TOU combinations (i.e. Generated, Consumed, etc.) |
| Service Type | The type of service being delivered at a Service Point (i.e. electric, gas, water). |


Copyright © 2026, Oracle and/or its affiliates.

| Term | Definition |
| --- | --- |
| Standard Time | This refers to a method of storing date/time information where there is no adjustment made for daylight savings time. A constant offset with Universal Coordinated Time (UTC) is maintained throughout the year. |
| Sub Usage Subscription | A "child" Usage Subscription that is subordinate to another Usage Subscription (such as when a utility is issuing a bill for a third party service). |
| Time Of Use (TOU) | A bucket of time throughout a day that's used for summarizing measurement data. These are often given a descriptive name as well (i.e. On Peak, Off Peak, etc.) |
| To Do | A work item that should be reviewed by a user. A To Do is related to data issues, incorrect processing, or other problems that occur within the system. |
| Unit Of Measure (UOM) | A metric of how measurement data is measured (i.e. kWh, kVARh, CCF, etc.) |
| Usage Calculation Group | A set of sequenced usage calculation rules used to calculate usage for a Usage Subscription. |
| Usage Calculation Rule | Business rules / logic used to calculate usage (bill determinants), such as a TOU-mapped consumption calculation. Each rule is a modular unit that can be grouped together and sequenced within a calculation group. |
| Usage Calculation Rule Eligibility Criteria | User-definable conditions that could cause a given usage calculation rule to be applied or skipped. This could involve the evaluation of some attribute of the device or measuring component, or some other important attribute. |
| Usage Subscription | A record of an ongoing request to send one or more Service Points' usage to one or more external systems (such as a billing application). |
| Usage Transaction | A record of bill determinant calculations for a Usage Subscription. |
| Usage Transaction Exception | An exception generated during the usage transaction process. Exceptions are assigned a severity that is used in determining whether or not the usage transaction should be transitioned into a requiring approval or resolution. |
| Validation, Estimation, and Editing (VEE) | The process by which initial measurement data is validated, estimated (if necessary) and edited (if necessary) based on a set of user-defined rules. |
| VEE Exception | An exception generated during Validation, Estimation and Editing (VEE) processing of initial measurement data. Exceptions are assigned a severity that is used in determining whether or not the initial measurement data should be transitioned into an exception state. |
| VEE Group | A collection of VEE Rules. |


Copyright © 2026, Oracle and/or its affiliates.

| Term | Definition |
| --- | --- |
| VEE Group Matrix (Factor) | A VEE Rule within a VEE group can be configured to pick from a list of VEE groups (referred to as a matrix) whose rules to execute next. This list of VEE groups is configured as the values of a factor. One example of its use could be to call geographically-specific VEE groups from within a larger-purpose group. A residential VEE group might contain<br>a rule that will pick the VEE group to execute based on Service Point location, where the VEE Group Matrix specifies:<br>Service Point in the North - VEE Group N<br>Service Point in the East - VEE Group E<br>Service Point in the South - VEE Group S |
| VEE Rule | Standard and custom Validation, Estimation and Editing (VEE) Rules that perform checking and/or manipulation of initial measurement data. |
| VEE Rule Eligibility Criteria | User-definable conditions that could cause a given VEE Rule to be applied or skipped. This could involve the evaluation of some attribute of the device or measuring component, or some other important attribute. |

## High Level Administration Areas

A high level overview of the administrative data used by Oracle Utilities Meter Data Management and other products has been defined in the Overview section of the *Oracle Utilities Meter Solution Administrative User Guide*. The methods behind how Oracle Utilities meter solution products are configured may be worth reviewing at some point to provide more context to the Business User Guide.

## Additional Resources

There are numerous ways to obtain additional information and resources to assist you when working with Oracle Utilities Meter solution products. The list of these options are maintained in the Additional Resources section of the *Oracle Utilities Meter Solution Administrative User Guide*.

# User Interface Features

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

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


Copyright © 2026, Oracle and/or its affiliates.

# User Interface Standard Features

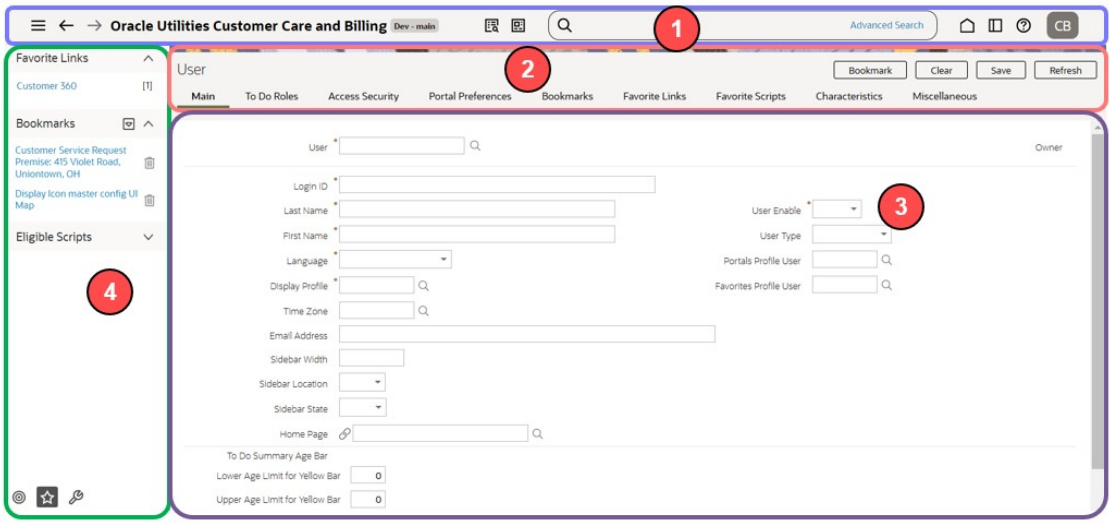
This section describes standard functionality realted to how the system looks and behaves.

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

## Page Components

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

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




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

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


Copyright © 2026, Oracle and/or its affiliates.

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

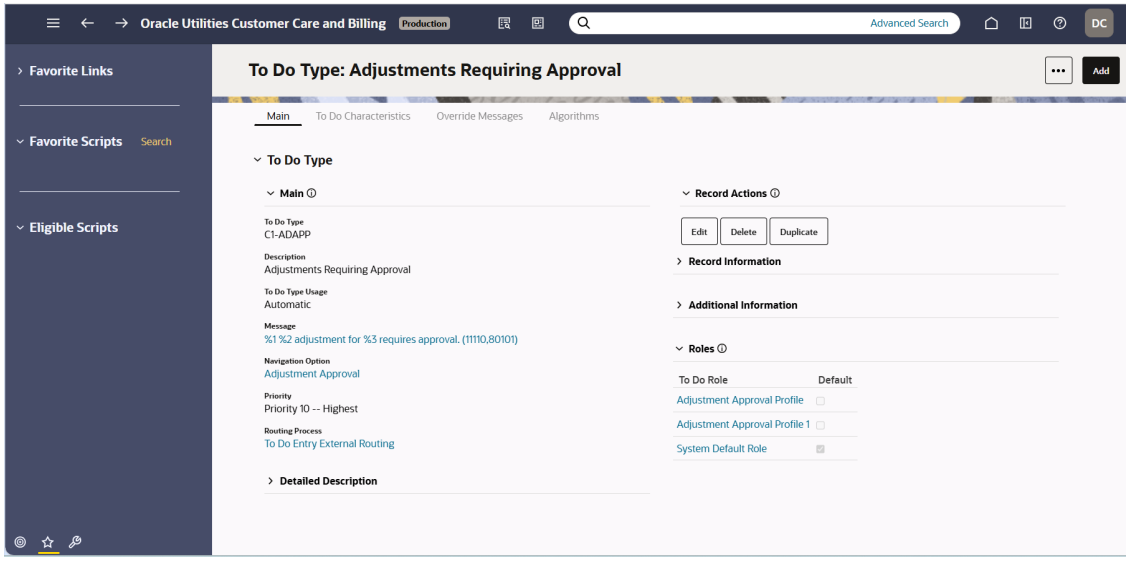
The next topics provide more information about each component.

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

## Redwood User Experience

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

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




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

- A colored theme is used for the application toolbar and the sidebar.
- Portal based pages follow the standard of the Label over the Field rather than before it.
- Some icons are different.


Copyright © 2026, Oracle and/or its affiliates.

- The expand / collapse control for a zone or section is moved to the left (or right for a right-to-left language).
- All input dialogues for portal based pages are displayed as a Drawer.
- When there are more than two page action buttons, one button is visible directly and the rest are grouped in an 'overflow' button (with three dots). This change applies to both portal based pages and fixed pages
- The spacing of elements is adjusted.

## The Application Toolbar

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

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

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

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

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

## Menu

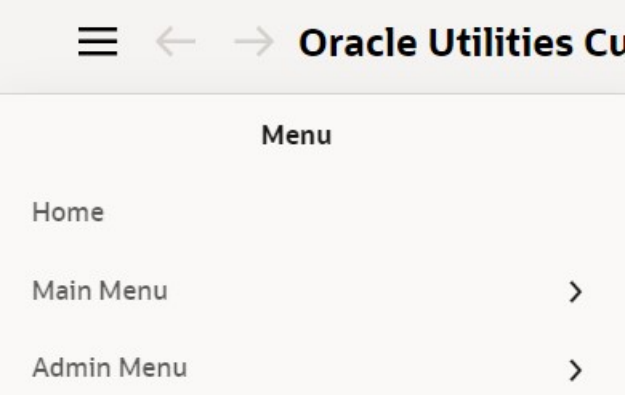
The Menu icon, , is available in the application toolbar to help you navigate to the different pages of the system.

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


Copyright © 2026, Oracle and/or its affiliates.

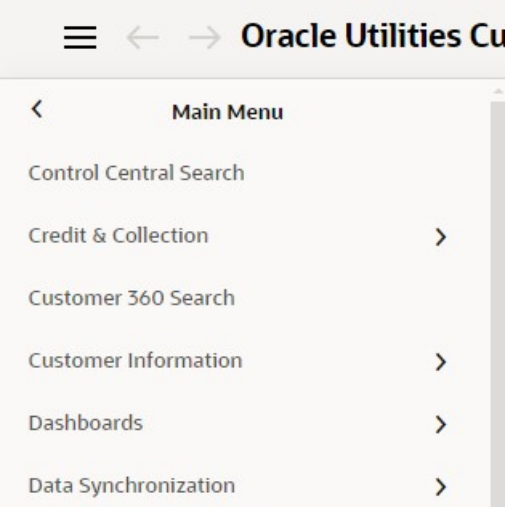
| Keyboard Shortcut |
| --- |
| Ctrl+Alt+M |

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



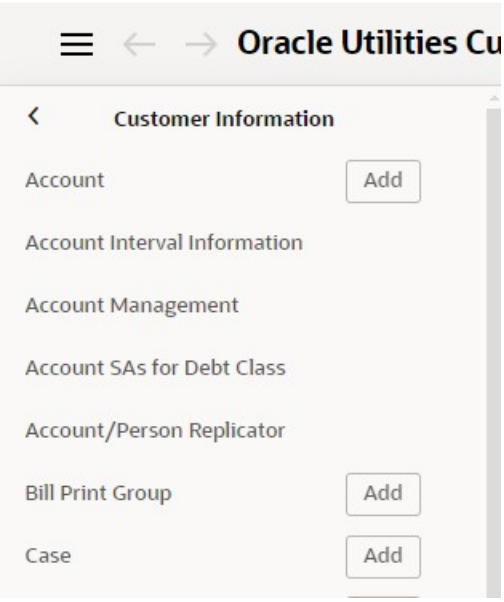

### Main Menu

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




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


Copyright © 2026, Oracle and/or its affiliates.




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

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

**Fixed:**

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

**Portal-Based:**

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

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

### Admin Menu

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

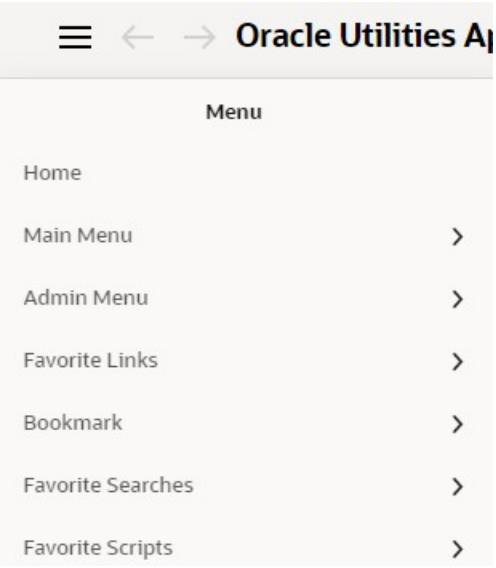
> **Note**
>
> Refer to Installation Options for information about the admin menu configuration options.


Copyright © 2026, Oracle and/or its affiliates.

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

**Favorites, Bookmarks**

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




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

## Back and Forward Arrows

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

**Back**

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

| Keyboard Shortcut |
| --- |
| `Alt+B` |

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

**Forward**

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

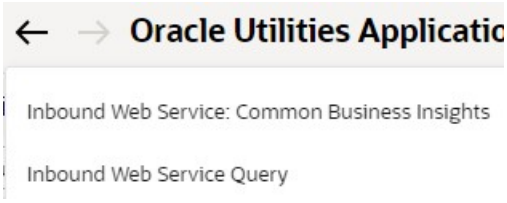

Copyright © 2026, Oracle and/or its affiliates.

| Keyboard Shortcut |
| --- |
| Alt+G |

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

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

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




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

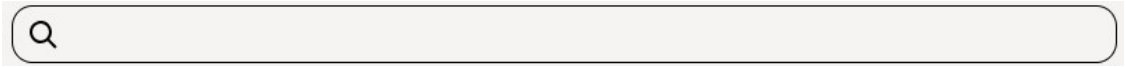
## Search

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

### Menu Item Search

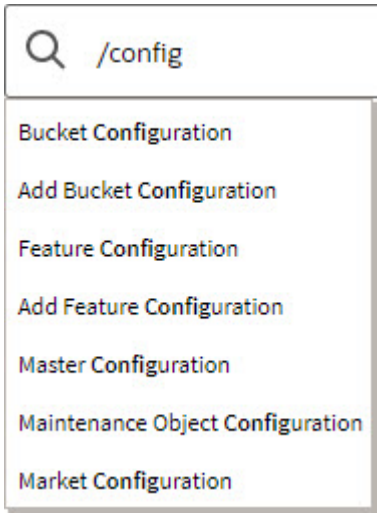
This search allows a user to enter the description of a menu item entry to navigate directly to the corresponding page or BPA script rather than using the menus to navigate to the desired page or script. If you start the search text with a slash "/", the search is confined to menu items. If you search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

| Keyboard Shortcut |
| --- |
| Ctrl+Alt+F |



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


Copyright © 2026, Oracle and/or its affiliates.

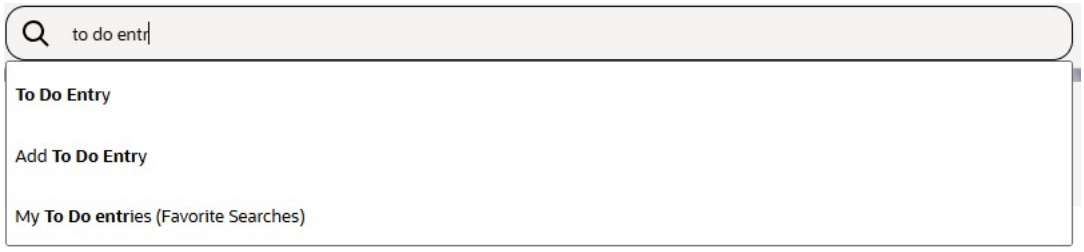



Note that for menu items within the Main Menu and Admin Menu, only menu entries that the user has security access for are included in the results. The text included in the search is taken from the menu item description. As described in Main Menu, menu lines often have options for search and Add. In most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

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

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

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



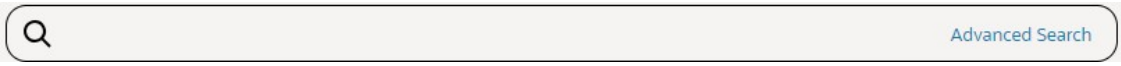

> **Note**
>
> The toolbar is responsive to window resizing or zooming in. The search field may be removed when the toolbar area is reduced.

### Unified Search

If your product has implemented unified search, you may use the toolbar search box to search for business entities. The unified search feature is a simplified version of your product's main search that allows you to lookup up records using free form text without leaving your current


Copyright © 2026, Oracle and/or its affiliates.

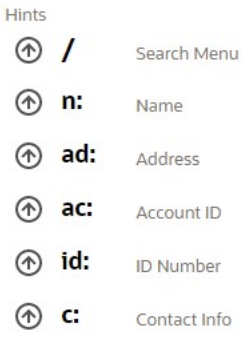
page. For more complex queries, use the link provided as part of the search box to navigate to your product's advanced search portal.




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

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

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




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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

## Home Icon

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

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

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

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

## Sidebar Toggle

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

**Table Sidebar icon list**

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

| Keyboard Shortcut |
|---|
| `Alt+J` |

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

## Help Menu

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

### Help

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

| Keyboard Shortcut |
| --- |
| `Alt+F1` |

### Help Assistant

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

### Prepare Issue Details

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

### Release Notes

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

### What's New

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

### Enable | Disable Debug

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

### About

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

The about window shows a variety of information including:

- The logged in user ID


Copyright © 2026, Oracle and/or its affiliates.

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

## User Menu

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

.

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

### Preferences

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

| Keyboard Shortcut |
| --- |
| `Ctrl+Alt+P` |

Refer to User Preferences for more information.

### Switch UI View

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

### Logout

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

| Keyboard Shortcut |
| --- |
| `Ctrl+Alt+L` |

# Page Title Area

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


Copyright © 2026, Oracle and/or its affiliates.

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

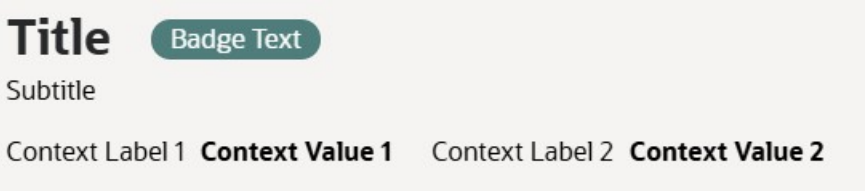
For certain portal pages in the system, the page title may have additional information.

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

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

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

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




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

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

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

- Page Actions Toolbar
- Portal Banners


Copyright © 2026, Oracle and/or its affiliates.

## Page Actions Toolbar

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

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

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

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

## Bookmark Button

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

| Keyboard Shortcut |
| --- |
| `Ctrl+Alt+B` |

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

## Next/Previous Item Buttons

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

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

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

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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

## Duplicate Button

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

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

| Keyboard Shortcut |
| --- |
| `Ctrl+Alt+D` |

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

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

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

## Delete Button

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

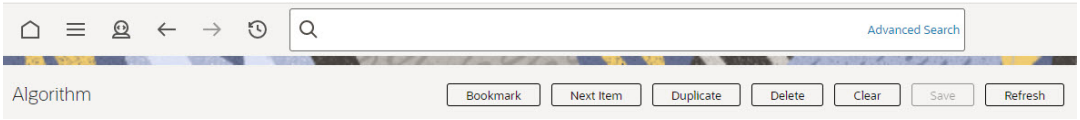
> **Note**
>
> If you make a change to an object and do not save the changes to the database, you cannot delete the object until you either save the changes or refresh the object with the contents on the database.


Copyright © 2026, Oracle and/or its affiliates.

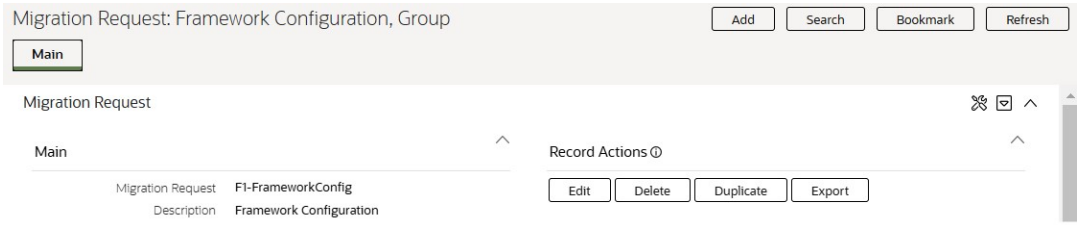
Before you delete an object, the system verifies that the object is not referenced anywhere in the system. If it is, you cannot delete it. If, for example, you attempt to delete a user, the system ensures this user has not been referenced in a user group or on a To Do.

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

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




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




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

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

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

## Clear Button

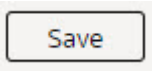
Click the **Clear** button to remove the values from the currently displayed object's fields. This action is typically used to clear input fields when adding a new object.

Copyright © 2026, Oracle and/or its affiliates.

| Keyboard Shortcut |
| --- |
| Ctrl+Alt+C |

## Save Button

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




| Keyboard Shortcut | Alternate Shortcut |
| --- | --- |
| Alt+S | Alt+Shift+S |

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

The button does not appear unless:

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

## Refresh Button

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

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

| Keyboard Shortcut |
| --- |
| Alt+R |

## Portal Maintenance

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

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

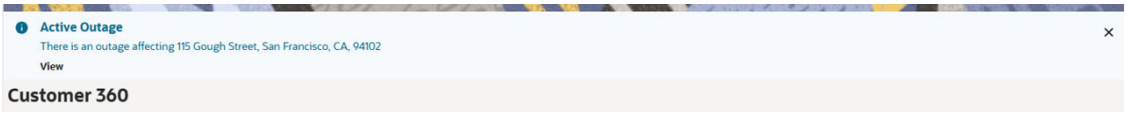
Use this action to navigate to the portal definition for the portal page you are currently viewing.


Copyright © 2026, Oracle and/or its affiliates.

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

## Portal Banners

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




The following describe important aspects of portal banner functionality:

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

# Object Display Area

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

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

- Multiple Tabs
- Drawer
- Text Box
- Date and Time Fields
- Input Field with Search
- Context Menu Button
- Go To Button
- Broadcast Button
- Embedded Help Icon
- Knowledge Article Icon
- Work Lists
- Scroll Bars with Dynamic Height
- Sorting Functionality

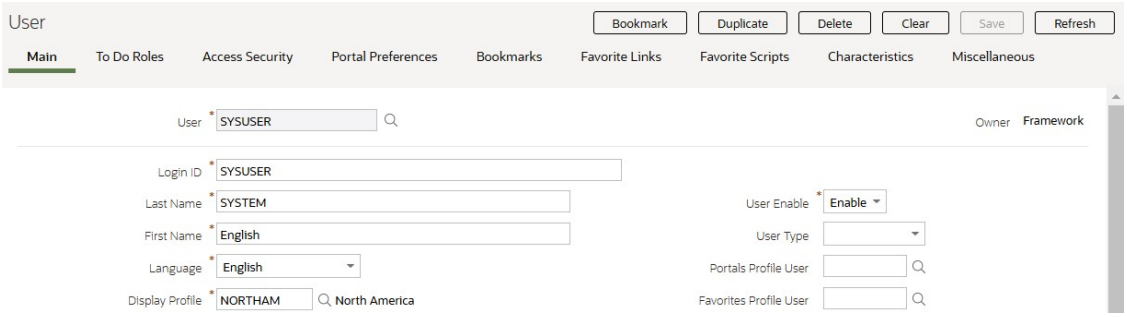

Copyright © 2026, Oracle and/or its affiliates.

- Get More Functionality
- Adding and Removing Rows Using Arrows
- Reorganizing Items Using the Up and Down Arrows
- Trees
- Required Field Indicator
- Accordion Windows

## Multiple Tabs

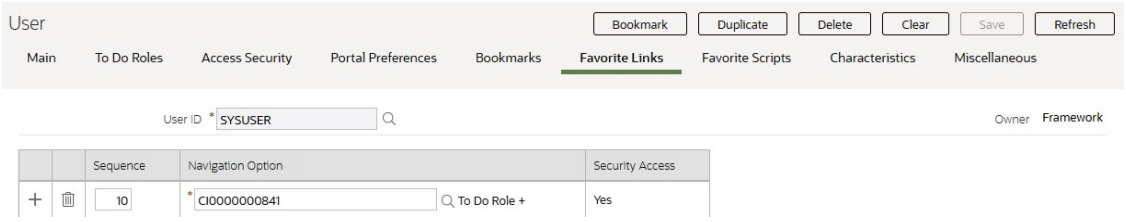
When a page in the application includes a lot of information, the page may be implemented with vertical scrolling to allow the user to view all the information. Or alternatively, the page may be implemented with several tabs, allowing the user to navigate between tabs to view or update information. The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

**Figure Multiple Tabs Example**




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

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




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


Copyright © 2026, Oracle and/or its affiliates.

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

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

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

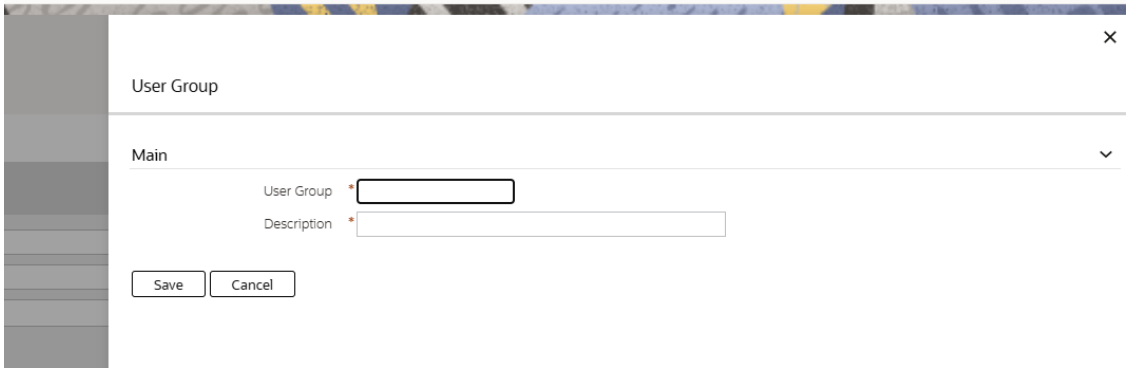
### Scoreboard Configuration

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

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

## Drawer

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





Copyright © 2026, Oracle and/or its affiliates.

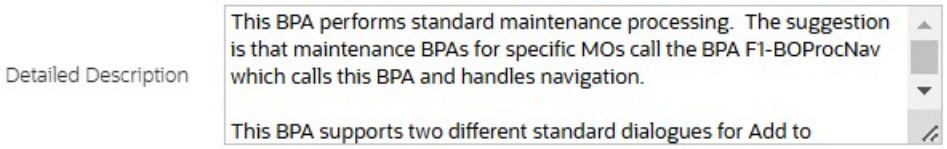
When applicable, a drawer may be launched from another drawer, forming a nested display of panels. You may navigate to the previous drawer using a 'back' action. Closing the top drawer closes all drawers.

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

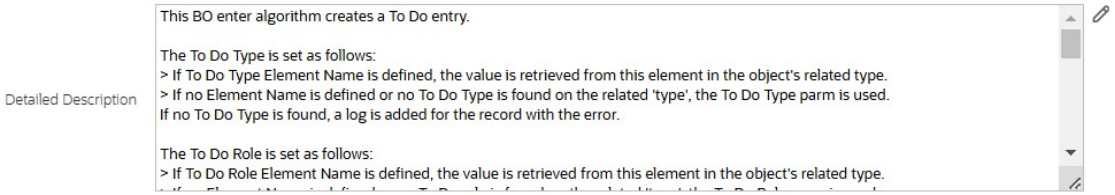
## Text Box

The following is an example of a basic text box:

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

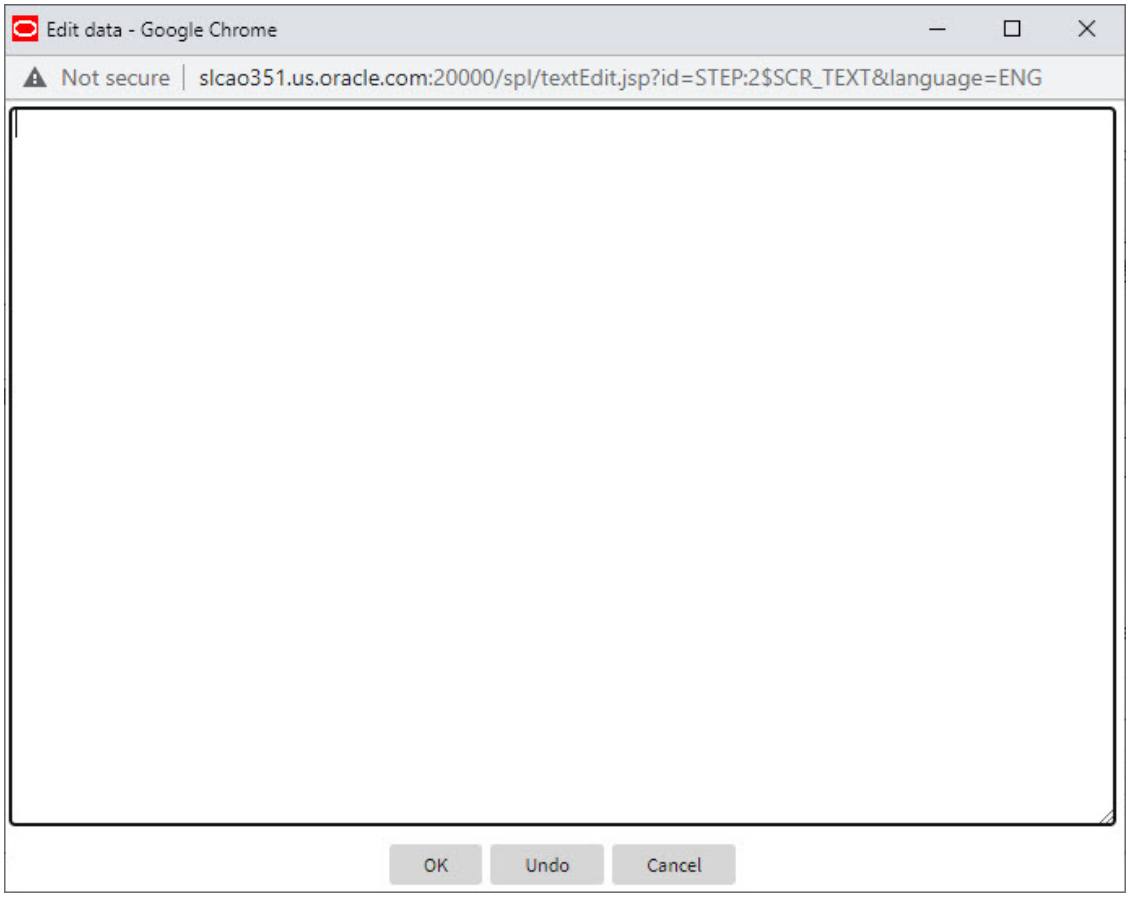



In our fixed pages, this type of text box also offers an expanded edit icon, (a pencil), that will appear adjacent to it.




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


Copyright © 2026, Oracle and/or its affiliates.




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


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

| Entered Character Description | Converted Character Description |
| --- | --- |
| Unicode urly left single quote. | Straight single quote. |


Copyright © 2026, Oracle and/or its affiliates.

| Entered Character Description | Converted Character Description |
| --- | --- |
| Unicode curly right single quote. | Straight single quote. |


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

| Entered Character | Unicode | Converted Character | Description |
| --- | --- | --- | --- |
| ellipse | U+2026 | … | Three dots. |


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

| Entered Character Description | Converted Character Description |
| --- | --- |
| U+00BC | 1/4 (using three separate character |


Copyright © 2026, Oracle and/or its affiliates.

| Entered Character Description | Converted Character Description |
| --- | --- |
| Unicode | s). |


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

| Unicode Entered Character Description | Converted Character Description |
| --- | --- |
| | s ) . |


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

| Entered Character Description | Unicode | Converted Character Description |
| --- | --- | --- |
| | | s ) . |

## Date and Time Fields

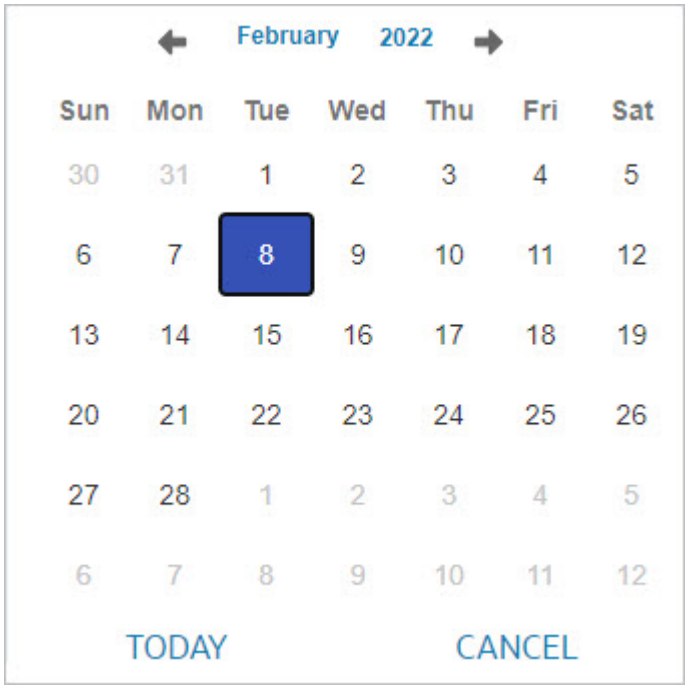
When both the date and time are required, the Date and Time fields typically look as follows:

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

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

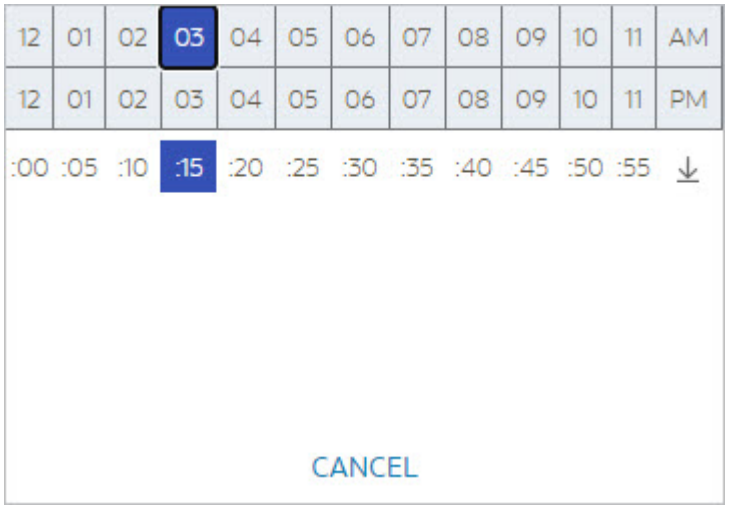
If you click on the calendar icon or press the Enter key while the cursor is positioned to the left of the icon in a Date or Date and Time field, the following window appears:


Copyright © 2026, Oracle and/or its affiliates.




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

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




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

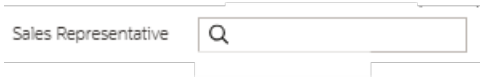
## Input Field with Search

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

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


Copyright © 2026, Oracle and/or its affiliates.

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




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

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

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

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

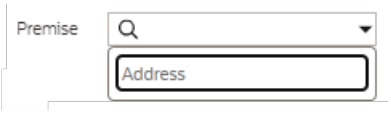
When an item is selected in the search page, the search page closes and the input field is populated with the value selected.

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

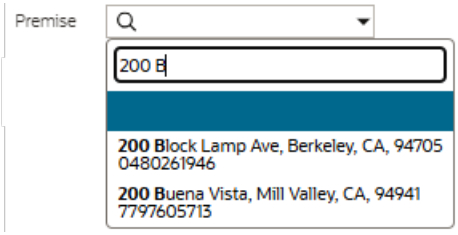
### Auto-filtering

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


Copyright © 2026, Oracle and/or its affiliates.




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

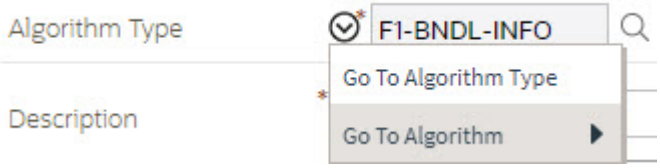



> **Note**
>
> Whether the text you type is matched only at the beginning of the string or at the middle of the string depends on how the filter is configured in the underlying search. Most search filters are configured to match text at the beginning of the string.

## Context Menu Button

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

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

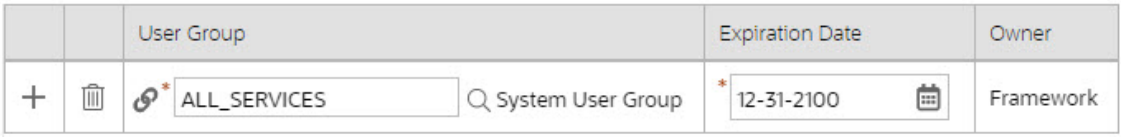



Different context menus exist for each of the major objects in the system; for example, in Customer Care and Billing, the context menu for Premise contains premise-oriented menu items, whereas the context menu for Account contains account-oriented menu items.

## Go To Button

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


Copyright © 2026, Oracle and/or its affiliates.




## Broadcast Button

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

## Embedded Help Icon

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

ⓘ

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

## Knowledge Article Icon

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

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

Refer to Knowledge Article for configuration information.

## Work Lists

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

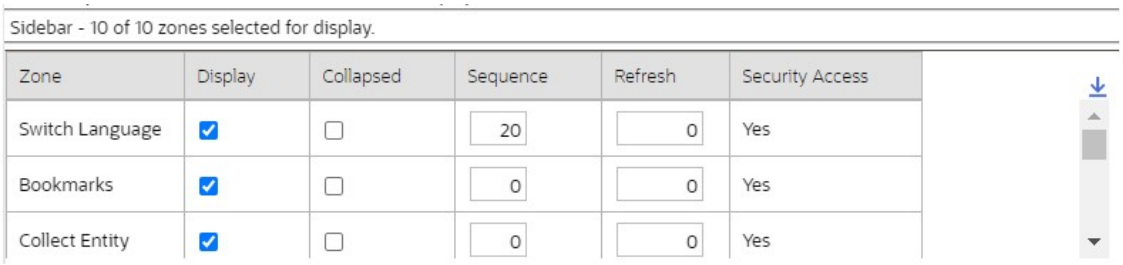
Clicking the icon results in populating the sidebar Work List Zone with the entire column. The user may use the Work List zone to easily process entries one at a time without returning back to the source query zone.

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

## Scroll Bars with Dynamic Height

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


Copyright © 2026, Oracle and/or its affiliates.




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

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

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

## Sorting Functionality

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

## Get More Functionality

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

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

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

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

## Adding and Removing Rows Using Arrows

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


Copyright © 2026, Oracle and/or its affiliates.

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

| | Sequence | Description |
|---|---|---|
| | 5 | Service Charge |

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

## Reorganizing Items Using the Up and Down Arrows

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

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

Notice the following about such a grid:

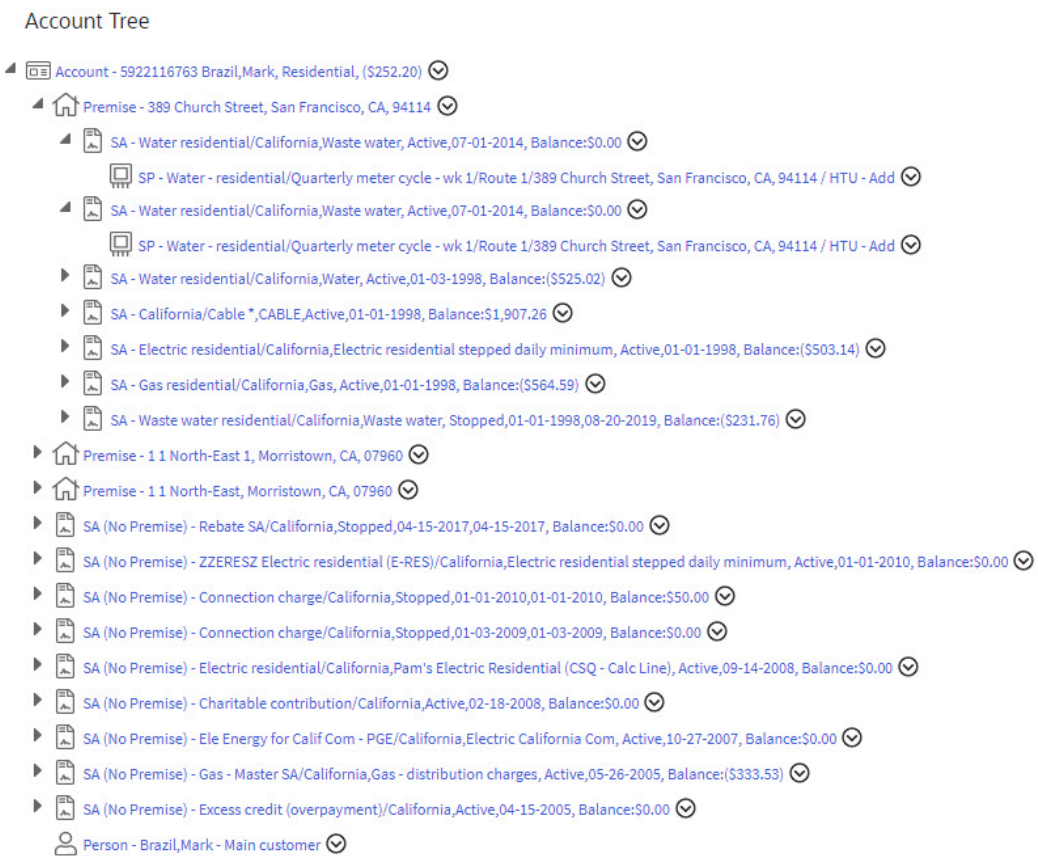
- The up arrow is dim in the first row. This is because it can't be moved further to the top.
- The down arrow is dim in the last row. This is because it can't be moved further down.
- The up and down arrows are usable in all other rows. Each time you click one of these arrows, the respective row is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

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

## Trees

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


Copyright © 2026, Oracle and/or its affiliates.




The following points describe how to use trees:

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

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

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

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

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

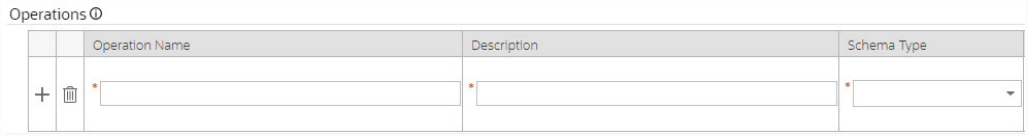
Nodes in trees fit into two broad categories:


Copyright © 2026, Oracle and/or its affiliates.

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

## Required Field Indicator

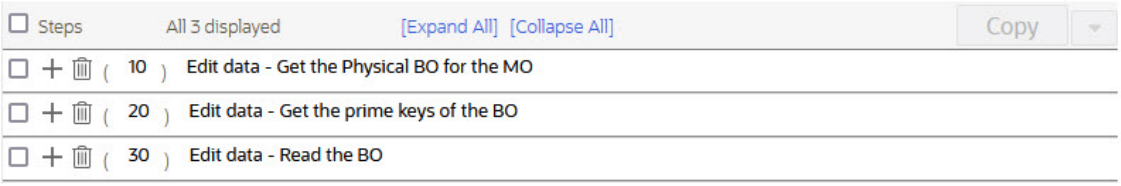
An asterisk (*) is used to indicate that a field is required. The system determines this based on the field's configuration. When a field becomes required because of an algorithm and not the field's configuration, however, the system will not add an asterisk.

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

## Accordion Windows

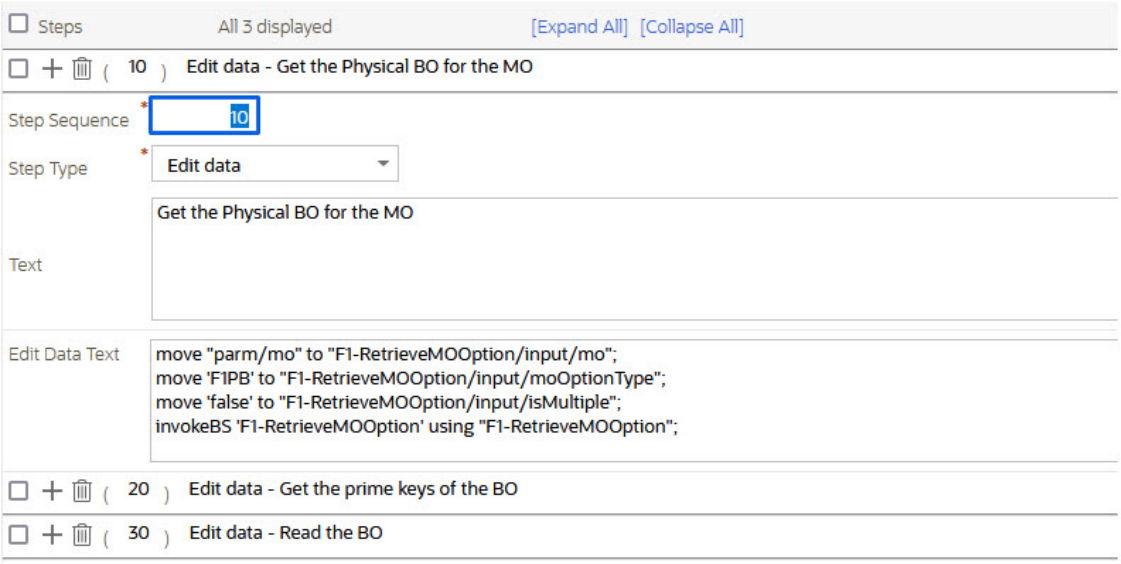
Accordion windows are used when the data held in an object's rows does not comfortably fit into a grid.

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




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


Copyright © 2026, Oracle and/or its affiliates.




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

## Sidebar Area

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

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

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

## Assistant Area

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


Copyright © 2026, Oracle and/or its affiliates.

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

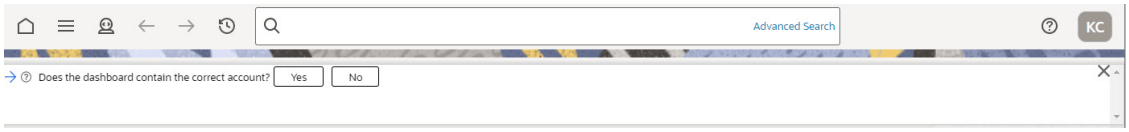
Refer to your product's specific documentation for more information about the functionality visible in this area.

## Script Area

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

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

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




# Portal Pages

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

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

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

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

## Portals Are Made Up Of Zones

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


Copyright © 2026, Oracle and/or its affiliates.

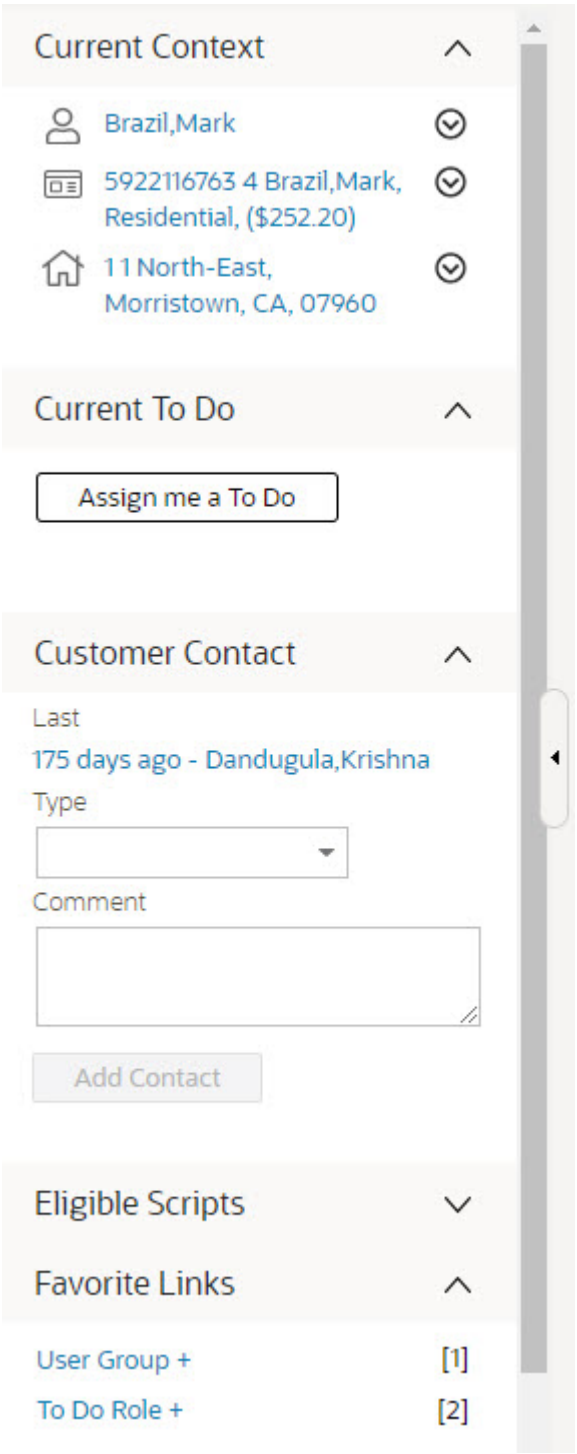
| Keyboard Shortcuts | Description |
| --- | --- |
| `Alt+]` | Moves the cursor focus to the next zone on the page based on the standard browser keyboard navigation "order". |
| `Shift+Alt+]` | Moves the cursor focus to the previous zone on the page based on the standard browser keyboard navigation "order".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

The following is a portal with seven zones:

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

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


Copyright © 2026, Oracle and/or its affiliates.




## Users Must Be Granted Security Access To Each Zone

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

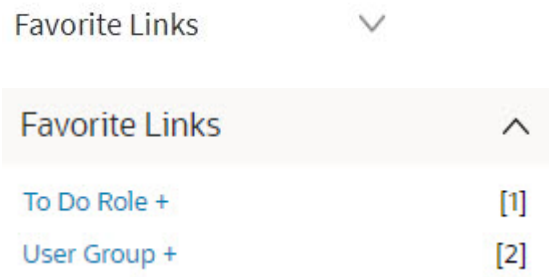
**NOT_SUPPORTED**

Refer to Granting Access to Zones for information .


Copyright © 2026, Oracle and/or its affiliates.

## Zones May Appear Collapsed When A Page Opens

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




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

## Some Portals Allow User Customization

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

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

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

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

## Sidebar Portal

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

## Bookmarks Zone

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

Click the trash can icon to delete the bookmark.

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

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


Copyright © 2026, Oracle and/or its affiliates.

## Collect Entity Zone

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

## Current To Do Zone

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

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

The following sections provide more detail about the zone.

### To Do Entry Visible

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

### Next To Do Assignment

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

### To Do Information

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

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

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

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

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

### Next / Previous Buttons

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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

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

### Related To Do Entries

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

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

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

### Implementations Not Using Substatus

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

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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

## Favorite Links Zone

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

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

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

## Favorite Scripts Zone

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

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

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

## Favorite Searches Zone

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

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

## Switch Language Zone

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

## Tags Zone

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

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

## To Do Summary Zone

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

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

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

## Work List Zone

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

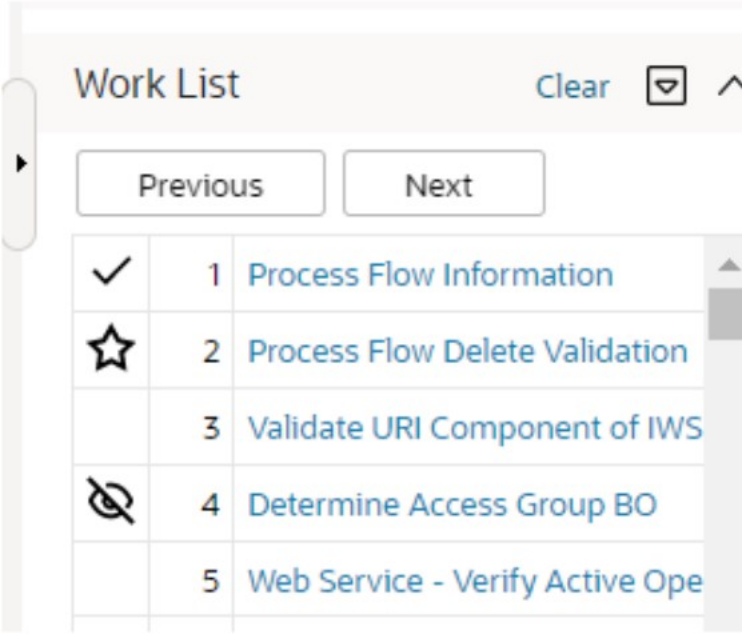
Previous and Next buttons are available in the zone allowing you to go to the next or previous entry without having to locate it yourself and click the link.

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

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

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


Copyright © 2026, Oracle and/or its affiliates.




## Zone Types and Characteristics

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

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

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

## Common Characteristics of Zones

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

- Zone Visibility

### Zone Visibility

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

For configuration information, refer to Zone Visibility Service Script.


Copyright © 2026, Oracle and/or its affiliates.

## Explorer Zones

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

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

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

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

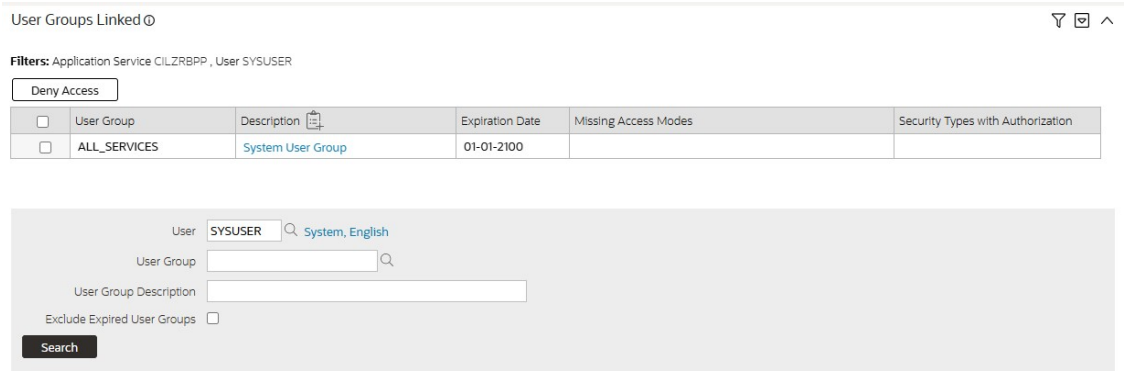
## Search Criteria / Filter Area

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

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

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

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





Copyright © 2026, Oracle and/or its affiliates.

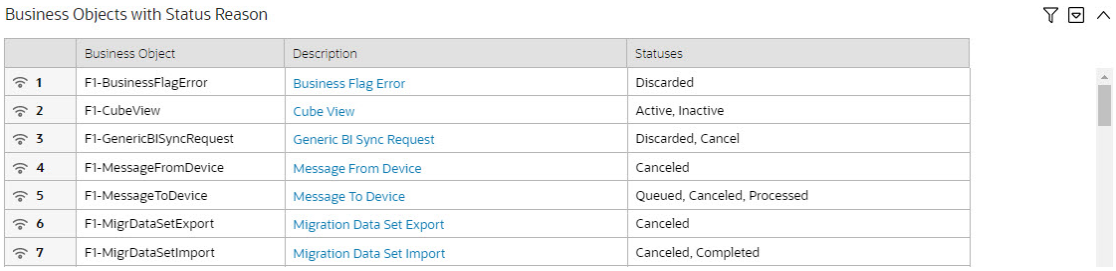
The info zone includes configuration to indicate whether the filter area, when exposed, should be shown at the top of the zone (above the results) or at the bottom of the zone (below the results). For query zones, the search criteria is always displayed at the top.

## Description Bar

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

## Results Area

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


Business Objects with Status Reason

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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

## Title Bar

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

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

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

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

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

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

### Explorer Zone Menu

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

- Clear Filters
- Export Data
- Save Search Options

### Clear Filters

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

### Export Data

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

Please note the following with respect to this feature:

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


Copyright © 2026, Oracle and/or its affiliates.

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

Save Search Options

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

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

Note the following additional points about this functionality:

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


Copyright © 2026, Oracle and/or its affiliates.

Save As

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

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

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

Open Saved Search

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

Save

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

Set As Default

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

Remove As Default

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

Add To Favorites

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

Remove From Favorites

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

Delete

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

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

Show Filters

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

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

Show Column Configuration Area

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

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

The Open Zone Maintenance Icon

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

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

## Configurable Columns

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


Copyright © 2026, Oracle and/or its affiliates.

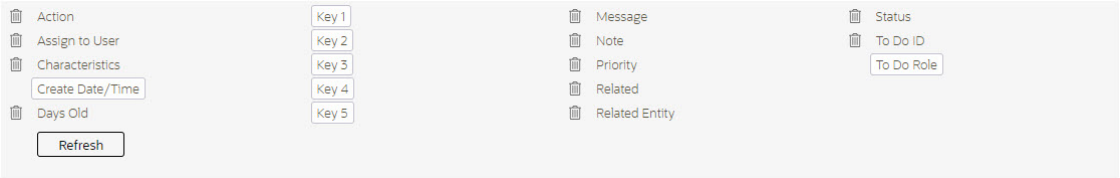
When a zones is configured to support this functionality, the Show Column Configuration Area icon is visible in the zone title area. Click this icon to expose the Column Configuration Area, if it is not visible and click the icon to hide the Column Configuration Area if desired.

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

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

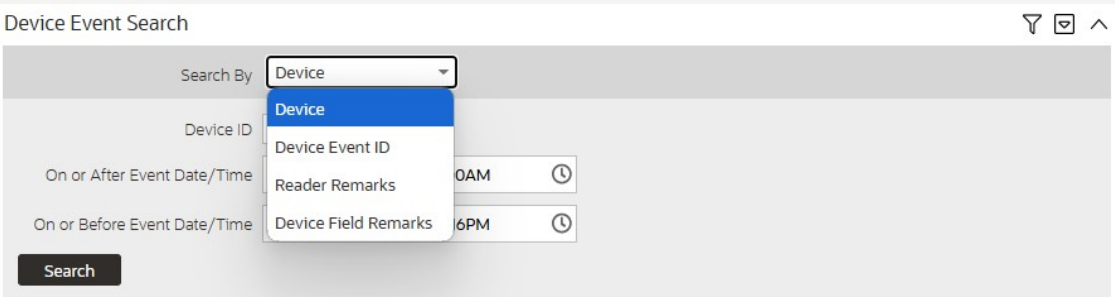
To remove a column that is currently in the results area, click the trash can icon next to the column name in the column configuration area and then click **Refresh** to rebuild the zone.

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




## Multi-Query Zones

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

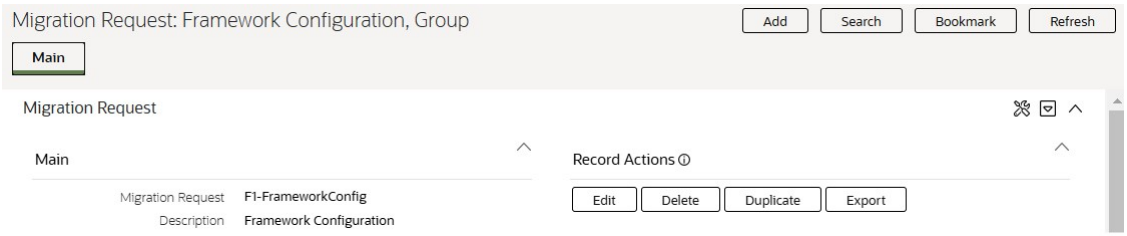



# Map Zones

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

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


Copyright © 2026, Oracle and/or its affiliates.




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

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

- Open Map Maintenance Icon

## Open Map Maintenance Icon

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

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

## Tabbed Display Zones

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

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

## Timeline Zone

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


Copyright © 2026, Oracle and/or its affiliates.

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

### Timelines Zones Are Configured By Your Implementation Team

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

### You Can Move Through Time

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

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

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

### Timelines Can Have Many Lines

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

### Each Line Shows Events

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

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

### Hovering Over An Event Shows the Event Information

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

- The event date.


Copyright © 2026, Oracle and/or its affiliates.

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

## Common Base Portals and Zones

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


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


### All-In-One Portal

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

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

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

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


- List Zone


### List Zone

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

The following functions are commonly available in a list zone:

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



Copyright © 2026, Oracle and/or its affiliates.

## Separate Query and Maintenance Portals

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

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

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

- Query Portal Search Zone
- Maintenance Portal

## Query Portal Search Zone

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

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

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

## Maintenance Portal

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

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

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

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

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

- Object Display Zone
- Log Zone

### Object Display Zone

This is a zone that displays the object being maintained.


Copyright © 2026, Oracle and/or its affiliates.

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

Log Zone

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

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

## Process Flow

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

Refer to Understanding Process Flows for more information.

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

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

# Fixed Pages

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

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

- Maintenance Pages
- Query Pages

## Maintenance Pages

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

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


Copyright © 2026, Oracle and/or its affiliates.

## The Unique Identifier Area

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

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

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

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

## Confirmation & Error Messages

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

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

If there are no errors, the system is modified.

## Warning If Unsaved Changes Exist

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

## Query Pages

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

- The Search Area

### The Search Area

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

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

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

## UI View Settings

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

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

# User Preferences

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

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


Copyright © 2026, Oracle and/or its affiliates.

  - Portal Preferences
  - Favorite Links
  - Favorite Scripts

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

# User - Main

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

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

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

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

**Description of Page**

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

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

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

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

**Display Profile** defines the display options for the user.

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

**Email Address** captures the user's email address.

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

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

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

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

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

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

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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

# User - To Do Roles

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

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

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

**Description of Page**

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

**Where Used**

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

# User - Access Security

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

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

**Description of Page**

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

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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

# User - Bookmarks

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

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

**Description of Page**

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

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

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

# User - Portal Preferences

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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

**Description of Page**

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

**Zone** lists all of the zones in the portal.

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

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

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

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

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

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

# User - Favorite Links

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

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

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

**Description of Page**


Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

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

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

# User - Favorite Scripts

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

**Description of Page**

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

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

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

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

# User - Characteristics

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

**Description of Page**

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

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

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

# User - Miscellaneous

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

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

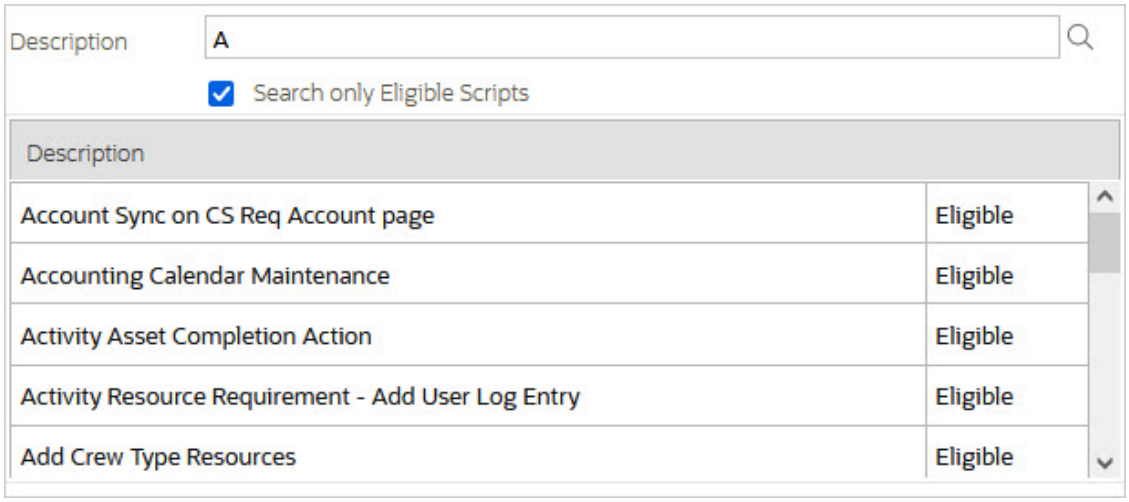

Copyright © 2026, Oracle and/or its affiliates.

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

# Initiating Scripts

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

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




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


Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

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

The following points highlight useful information common to all scripts:

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

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

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

# Shortcut Key Summary

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


Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

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


Copyright © 2026, Oracle and/or its affiliates.

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

# Asset Management


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The asset management functionality described in this section is available with the following products:
>
> - Oracle Utilities Advanced Meter Solution (C2M-AMS)
> - Oracle Utilities Customer to Meter (C2M)
> - Oracle Utilities Customer Cloud Service (CCS)
> - Oracle Utilities Meter Solution Cloud Service (MSCS)


- Assets
- Asset Activities
- Asset Activity Generators
- Asset Completion Events
- Asset Inbound Communications
- Asset Locations
- Asset Replication
- Manufacturers
- Mass Asset Disposition Update
- Service History
- Shipping Memos
- Specifications


# Assets


- About Assets
- Maintaining Assets


## About Assets

An asset describes such objects as meters, poles, pipes, transformers, components, or any other material item owned or managed by an organization. Stock items that are "Tracked" can be linked to the Asset record when the tracked stock item defined on the Specification record is associated with the Asset record. A tracked stock item defined as hazardous material will also display within the **Maintenance** tab on the asset record.

Key aspects of an asset are as follows:

- Have monetary or functional value
- Able to be inspected
- Able to be serviced and maintained
- Includes a disposition history
- Could include one or more components



Copyright © 2026, Oracle and/or its affiliates.

- Includes a unique asset identifier

Both the Asset Number and Badge Number identified for a particular asset must be unique within the system. These numbers can be entered into the system manually or an administrator can configure individual asset types to automatically generate the asset number and/or badge number for an asset created from that asset type. Please refer to *Understanding Asset Types* for information on how to configure the system for automatic generation of asset and badge numbers.

Assets are always associated with a location. Refer to About Asset Locations for more information.

Assets typically exist within a structured hierarchy based on the relationship between the asset and its locations and organizations.

### Fixed Assets

Fixed assets (or Capital assets) are generally company-owned long-term assets; having a record in the asset table with an asset ID. These assets are initially capitalized but then depreciate over time. All fixed assets must be associated with a cost since cost is part of the value of the fixed assets for depreciation purposes. Deprecation is currently not managed within **Oracle Utilities Work and Asset Management** however, **Oracle Utilities Work and Asset Management** can interface fixed assets, for example, property unit, quantity, in service date, value, and so on, to an external fixed asset management system. Fixed asset information is most often obtained during the receipt and accept process or through the construction work activity process however, there is an exception for plant based capital work. Plant based capital work is unique because it usually involves standalone assets that are very large and expensive and obtaining information on these assets fall outside of these processes, so a secured manual process can be used. This manual process enables users to populate specific fields on the **Asset Performance** tab, such as *Member Asset*, *Fixed Asset Number*, *Additional Cost*, *ERP Project Number*, and *ERP Subproject Task Number*, and send this information to the external fixed asset system using a secured **Send Asset Information Update** button.

To use this feature, implement the following:

- Define your Property Units (Groupings of fixed assets).
- Identify which **Work and Asset Management** assets are considered "Member Assets" (These are fixed assets that are not individually depreciated such as poles, conductors, or pipes).
- Ensure the ERP Project and Task Numbers are known (This is needed if there is a Project Integration. Sends specific fixed Asset information to ERP Project instead of Fixed Assets system).
- Add two new Access Modes to the appropriate User Groups (Update Fixed Asset Information (W1UF) and Send Fixed Asset Information (W1SF).
- Ensure the application services for Fleet Asset, General Asset, Infrastructure Asset, and Smart Meter have these access modes (Users that need access to this feature must have these application services and access modes).

### Linear Assets

Setting up your assets so that they are recognized as a “system” can help you to manage maintenance. If your organization uses GIS, this functionality can also help to position your assets on the mapping system that you have in use. Please consider the following when configuring linear assets:

- The Asset Location Type and Asset Location determine where the linear asset is connected.


Copyright © 2026, Oracle and/or its affiliates.

- Linear assets include a “Connected to” field which allows you to define the connections between assets

  For example, a pipe system would consist of a pipe, connected to a manhole, connected to the next pipe, and so on.

# Maintaining Assets

You use the **Asset** portal is used to display and maintain assets.

Refer to About Assets for additional functional information.

You can access the portal from **Menu > Asset Management > Asset**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset**: This zone displays the basic attributes of the asset.
- **Attached Components**: This zone displays a list of any components that are installed on the asset.
- **Disposition History**: This zone displays a log of the status and location changes for the life of the asset.

The following zones may appear as part of the portal's **Device** tab page:

- **Device**: This zone defines the basic attributes of the device
- **Device Configuration List**: This zone lists the device configurations for the device
- **Device Activities**: This zone lists activities related to the device in date descending order
- **Device Events**: This zone lists the most recent device events associated to the device in descending order
- **Device Exceptions**: This zone ists the first 50 VEE exceptions for all measuring components for the device, sorted by the exception creation date in descending order

The following zones may appear as part of the portal's **Maintenance** tab page:

- **Asset Service History**: This zone displays a log of the service history that has been recorded for the life of the asset.
- **Activity History**: This zones displays a log of the activities that have been performed on the asset over the life of the asset.

The following zones may appear as part of the portal's **Tree View** tab page:

- **Asset Hierarchy**: This zone displays the service point at which the asset is currently installed (if applicable). Clicking the broadcast icon opens the **Installed Assets** zone, which displays a tree view of all of the assets installed at the service point.
- **Currently Installed Assets**: This zone displays a list of all assets currently installed in this location.
- **Installed Assets**: This zone displays a tree view of all of the assets installed at the service point.

The following zones may appear as part of the portal's **Log** tab page:

- **Asset Log**: This zone displays the server logs for the asset.


Copyright © 2026, Oracle and/or its affiliates.

- **Asset Sync Requests**: This zone displays a list of synchronization requests related to the asset. Synchronization requests are used to synchronize the asset with an associated device.

> **Note**
>
> The **Asset** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

### Working with Assets

A number of actions are available for an asset depending on its status:

| Action | Allowed Statuses | Details |
|---|---|---|
| Store | In Receipt<br>Installed<br>In Repair<br>Retired | Moves the asset to a storage (“out of service”) location and sets the asset’s status to “In Store”.. |
| Reject | In Receipt<br>In Store | Sets the status of the asset to “Rejected” (used if an asset fails an inspection) |
| Install | In Store | Installs the asset at an asset location (service point) and sets the asset’s status to “Installed”. |
| Repair | In Store<br>Retired | Sets the status of the asset to “In Repair”. |
| Retire | In Store<br>In Repair | Sets the status of the asset to “Retired”. |
| Pending Disposition | In Store<br>Installed<br>In Repair<br>Retired | Sets the status of the asset to “Pending Disposition”. |
| Change Location | In Store | Changes the storage location of the asset. |

> **Note**
>
> If an asset is deleted, its related device is also deleted.

# Asset Activities


- About Asset Activities
- Maintaining Asset Activities


## About Asset Activities



Copyright © 2026, Oracle and/or its affiliates.

Asset activities are used for creating field work tasks related to an asset or asset location, such as installation or removal of a meter to and from its service point. It may also be used for creating field work tasks for a specific location unrelated to an asset such as trimming trees near power lines, fixing a street light pole, repairing fire hydrant leaks, and so on.

The base package includes an asset activity type business object (W1-SOMActivityType) and a corresponding asset activity business object (W1-SOMActivity), used in the integration or consolidation of Operational Device Management activities with and Service Order Management activities related to field work.

When an asset activity (based on the asset activity type and asset activity business objects referenced above) is created and enters the “Communication in Progress” state, the activity is synchronized with Service Order Management, creating a service order field activity with a field task type that corresponds to the asset activity’s type.

Once the field activity has been created, Service Order Management is responsible for dispatching the request to the field work system. When the task is completed, Service Order Management receives the completion information from the field work system, completes the service order field activity and sends the completion information to Operational Device Management to complete the corresponding asset activity.

Asset activities (based on the (W1-SOMActivity business object) are defined by:

- **Status**: The current status of the asset activity
- **Emergency**: A flag that indicates if the asset activity is an emergency or nor
- **Originating Activity**: The activity that triggered the creation of the asset activity (if applicable)
- **Location**: The location where the asset activity is to be performed
- **Asset**: The asset associated with the asset activity
- **Activity Generator**: The asset activity generator that created the asset activity (if applicable)
- **Service Date/Time**: The date and time at which the field work associated with the asset activity is to be performed
- **Address Information**: The address where the field work associated with the asset activity is to be performed
- **Contact Information**: Contact information (customer name and phone number) related to the asset activity

# Maintaining Asset Activities

You use the **Field Activity Query** and **Field Activity** portals to display and maintain asset activities.

Refer to About Asset Activities for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Activity**. You are brought to the **Field Activity Query** portal with options for searching.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity**: This zone displays the basic attributes of the asset activity.
- **Completion Information Entry**: This zone displays a list of completion events that must be processing to finalize the asset activity


Copyright © 2026, Oracle and/or its affiliates.

- **Attachments**: This zone displays a list of attachments (if any) related to the asset activity.

The following zones may appear as part of the portal's **Log** tab page:

- **Activity Log**: This zone displays the server logs for the asset activity.

> **Note**
>
> The **Field Activity** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

> **Note**
>
> For asset activities that have been synchronized with Service Order Management, you are brought to the **Activity** portal to view and maintain the selected record. See About Service Order Field Activities for more information.

# Asset Activity Generators


- About Asset Activity Generators
- Maintaining Asset Activity Generators


## About Asset Activity Generators

Asset activity generators are used for mass generation of activities. An Asset Generator Criteria business object is selected when creating asset activity generators, and is used for entering the criteria values for searching assets and asset locations that will be linked to the activities to be created. Such criteria includes location type, asset identifiers, location identifiers, address components, asset disposition, indication if asset is installed or not, asset type, specification and among others. Required information also includes, the activity type, disposition of the created activities, maximum number of activities that will be created, and an indication if the activities will be created via batch processing.

The base package the following set of activity generator business objects containing the various criteria:

- **Asset Criteria** (W1–AssetCriteriaActGenerator): used to define asset-related criteria used to select the assets for which activities will be created (for example, activities for all assets for a certain asset type, a specified current disposition or condition, or whose identifiers contain specific characters, and so on)
- **Asset List** (W1–AssetListActGenerator): used to define a list of assets for which activities will be created
- **Component Criteria** (W1–CompnCriteriaActGenerator): used to define component-related criteria to select the components for which activities will be created (for example, asset inspection activities for all assets for a certain component type, a specified current disposition or condition, or whose identifiers contain specific characters, and so on)
- **Component List** (W1–CompnAssetListActGenerator): used to define a list of components for which activities will be created



Copyright © 2026, Oracle and/or its affiliates.

- **Location Criteria** (W1–NodeCriteriaActGenerator): used to define location-related criteria to select the asset locations for which activities will be created (for example, activities for all locations where assets are current installed, locations of a specific type, or whose identifiers contain specific characters, and so on)
-

Asset activity generators are defined by:

- **Status**: The current status of the asset activity generator
- **Batch Generation Requested**: A flag that indicates if the activities created by the asset activity generator should be generated via batch processing
- **Generator Information**: Details related to the specific activity generator business object (Asset List, Asset, Location, etc.) used by the asset activity generator
- **Activity Information**: Details for the types of activities created by the asset activity generator, including activity type, default disposition, activation and work lead days (defaulted from the asset activity type), and the maximum number of activities to be created

Specific asset activity generators contain additional information based on their business object. For instance, asset activity generators based on the Location Criteria business object includes **Location Information** and **Address Information** sections.

# Maintaining Asset Activity Generators

You use the **Activity Generator** portal is used to display and maintain asset activity generators.

Refer to About Asset Activity Generators for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Activity Generator**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Generator**: This zone displays the basic attributes of the asset activity generator.
- **Activity Information**: This zone displays summary information about the activities created by the activity generator.

The following zones may appear as part of the portal's **Activities** tab page:

- **Activities**: This zones displays a list of the activities created by the activity generator.

The following zones may appear as part of the portal's **Log** tab page:

- **Activity Generator Log**: This zone displays the server logs for the asset activity generator.

> **Note**
>
> The **Activity Generator** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.


Copyright © 2026, Oracle and/or its affiliates.

# Asset Completion Events


- About Asset Completion Events
- Maintaining Asset Completion Events


## About Asset Completion Events

Asset completion events are used to execute business events within the application. For example, completion events may be configured for several types of asset processing such as asset update, state transition, replication, service history and activity creation.

These are created automatically by the inbound communication through the completion event types configured on the activity type of the activity that the inbound communication is attempting to complete.

> **Note**
>
> Since these records are most often created by the system, most users will only rarely be manually creating or updating asset completion events.

Asset completion events are defined by:

- **Status**: The current status of the asset completion event
- **Execution Sequence**: The sequence number for the asset completion event
- **Inbound Communication**: The asset inbound communication that created the asset completion event
- **Relationship Information**: The asset, component, location, or service history associated with the asset inbound
- **Event Information**: Information about the asset completion event, including date and time of completion and any comments

## Maintaining Asset Completion Events

You use the **Completion Event** portal is used to display and maintain asset completion events.

Refer to About Asset Completion Events for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Completion Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Completion Event**: This zone displays the basic attributes of the asset completion event.
- **Completion Event Related Objects**: This zone displays objects related to the asset completion event, such as a device, service point, or initial measurement.



Copyright © 2026, Oracle and/or its affiliates.

- **Attachments**: This zone displays attachments, if any, related to the asset completion event.

The following zones may appear as part of the portal's **Log** tab page:

- **Completion Event Log**: This zone displays the server logs for the asset completion event.

> **Note**
>
> The **Completion Event** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

# Asset Inbound Communications


- About Asset Inbound Communications
- Maintaining Asset Inbound Communications


## About Asset Inbound Communications

Asset inbound communications are used to stage the completion of an asset activity. This is typically created automatically by the system following the completion of the corresponding field activity in the field work system (such as Oracle Field Service Cloud).

> **Note**
>
> Since these records are most often created by the system, most users will only rarely be manually creating or updating asset inbound communications.

Asset inbound communications are defined by:

- **Status**: The current status of the asset inbound communication
- **Initiating Activity**: The asset activity, typically a field activity, that triggered the creation of the asset inbound communication
- **Event Information**: Information about the completion event associated with the asset inbound communication, including date and time of completion and any comments
- **Relationship Information**: The asset, component, location, or service history associated with the asset inbound communication
- **Asset Identifier Information**: Identifier types and values used to establish a connection between an asset and a corresponding entity in an external system
- **Attached to Asset Identifier Information**: Identifier types and values used to establish a connection between a component and related asset, and a corresponding entity in an external system
- **Location Identifier Information**: Identifier types and values used to establish a connection between a location and a corresponding entity in an external system
- **Activity Type Identifier Information**: Identifier types and values used to establish a connection between an asset activity type and a corresponding entity in an external system



Copyright © 2026, Oracle and/or its affiliates.

## Maintaining Asset Inbound Communications

You use the **Inbound Communication** portal is used to display and maintain asset inbound communications.

Refer to About Asset Inbound Communications for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Asset Inbound Communication**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Inbound Communication**: This zone displays information for the asset inbound communication.
- **Communication Tree**: This zone displays a tree of objects related to the asset inbound communication, including its parent activity.

The following zones may appear as part of the portal's **Log** tab page:

- **Inbound Communication Log**: This zone displays the server logs for the asset completion event.

> **Note**
>
> The **Inbound Communication** portal may display tabs and zones beyond those described here. These tabs and zones are not used with Oracle Utilities Customer To Meter.

# Asset Locations


- About Asset Locations
- Maintaining Asset Locations


## About Asset Locations

Asset locations define the physical location where an asset is installed. In Oracle Utilities Customer To Meter, asset locations are service points, and specify the address where a device is installed.

In Oracle Utilities Customer To Meter, asset locations are created when a service point is created, and are maintained using the **Service Point** portal. If the service point has a contact associated with it, a corresponding contact will be created for the asset location. See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with asset locations.

Note that an asset location is the physical location where assets or components can be installed, whereas Out of Service Locations are for receipt, storage, repair, retirement, transport (on vehicles) and other dispositions.



Copyright © 2026, Oracle and/or its affiliates.

## Maintaining Asset Locations

You use the **Asset Location** portal is used to search for asset locations.

Refer to About Asset Locations for additional information about asset locations.

You can access the **Asset Location** portal from **Menu > Asset Management > Asset Location**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the **Service Point** portal to view and maintain the selected record.

> **Note**
>
> In Oracle Utilities Customer To Meter, asset locations are created and/or updated when a service point is created and/or updated. If the service point has a contact associated with it, a corresponding contact will be created for the asset location. See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with asset locations.

# Asset Replication

- About Asset Replication
- Maintaining Asset Replication

## About Asset Replication

As devices and other assets are received in large quantities, the **Asset Replication** portal allows for the receipt, inspection, storage and other processing of these assets in batches.

Using asset replication functionality your organization can complete processes such as:

- Replicating assets in the same lot/pallet
- Selecting assets for testing or inspection
- Mass updating asset disposition such as rejecting or accepting a lot or pallet

This processing creates new asset IDs, badge numbers and serial numbers for assets based on general information that is entered related to each batch. When a new replication event is created it is tied to a replication activity type to determine default settings and attributes for the replication. The actual asset information can be entered manually or uploaded via a CSV file.

Assets can then be moved between locations, or dispositioned using Mass Asset Disposition Update.

## Maintaining Asset Replication

You use the **Asset Replication** portal is used to display and maintain asset replication activities.

Refer to About Asset Replication for additional functional information.


Copyright © 2026, Oracle and/or its affiliates.

You can access the portal from **Menu > Asset Management > Asset Replication**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset Replication**: This zone displays the basic attributes of the asset replication activity.
- **Completion Event**: This zone displays a list of completion events related to the asset replication activity.

# Manufacturers


- About Manufacturers
- Maintaining Manufacturers


## About Manufacturers

Manufacturers are the companies that makes assets (devices, communication components, transformers, etc.).

An asset is related to a manufacturer through the asset's specification.

Each manufacturer can have zero or more models defined. Models for a single manufacturer can have diverse service types.

## Maintaining Manufacturers

You use the **Manufacturer** portal is used to display and maintain manufacturers.

Refer to About Manufacturers for additional functional information.

You can access the portal from **Menu**, then **Asset Management**, then **Manufacturer**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Manufacturer**: This zone displays the basic attributes of the manufacturer.
- **Manufacturer Models**: This zone displays a list of models associated with the manufacture. To add or edit models available for the manufacturer, click **Edit** in the zone title bar.
- **Manufacturer Specifications**: This zone displays a list of specifications associated with the manufacturer. Use the **Specification** portal to add or edit the specifications associated with the manufacturer.

# Mass Asset Disposition Update

You use the **Mass Asset Disposition Update** portal is used to update the disposition of a group of assets.



Copyright © 2026, Oracle and/or its affiliates.

More specifically, Mass Asset Disposition is used for moving many assets of the same type from one location to another location. In a mass disposition task, the target is always a single asset location. For example, if a pallet of meters has been received in the receiving asset location, mass disposition can be used to move this batch of assets to a storage asset location or an out of service location.

Refer to About Assets for additional functional information.

To use the **Mass Asset Disposition Update** portal:

1. Select **Menu > Asset Management > Mass Asset Disposition Update**.
   You are brought to a query portal with options for searching assets
2. Enter search criteria appropriate to the asset you wish to update and click **Search**.
3. From the list of search results, select the assets you wish to update and click **Change Disposition**.The **Mass Asset Disposition Update** screen opens.
4. Enter the date and time at which the update should take effect in the **Effective Date/Time** field.
5. Select the new disposition for the assets from the **Asset Disposition** drop-down list.
6. Select the new location for the assets from the **Location** drop-down list or field.
7. Select or enter other data as appropriate (based on the new disposition). For instance, if you select “Rejected” you must also select a reason from the **Reason** drop-down list.

# Service History

- About Service History
- Maintaining Service History

## About Service History

Service history is used to capture and record relevant information regarding service or maintenance on assets. Typically it is used to record inspection feedback, pass/fail details, downtime, parts failure information, maintenance or service logs, or other information regarding service on the asset.

Types of service history might include:

- Asset downtime
- Failure details
- Asset maintenance
- Visual inspection
- Leak detection
- Smart meter test
- Battery replacement
- ..., and so on.

Service history records can be manually created or automatically generated as the result of completion events being processed for activities. If the service history type has any schedule defined, a set schedule may also be established to create regular service history for repeated


Copyright © 2026, Oracle and/or its affiliates.

tasks such as inspection or regular maintenance. Service history types linked to asset types control which service history can be created for assets.

## Maintaining Service History

You use the **Service History** portal is used to display and maintain service history records.

Refer to About Service History for additional functional information.

You can access the portal from **Menu > Asset Management > Service History**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service History**: This zone displays the basic attributes of the service history record.

# Shipping Memos


- About Shipping Memos
- Maintaining Shipping Memos


## About Shipping Memos

Shipping memos are records of stock items being shipped from an internal storeroom, whether from storeroom to storeroom or from storeroom to vendor.

Shipping memos can be created for both standard and tracked stock items and enables storeroom managers and shipping clerks a way to manage and track stock items being shipped between locations.

Some uses for shipping memo include the following:

- Tracking stock items returned to vendors for credit, replacement, or repair
- Helping to manage the return of repaired or replacement stock items back into inventory

Oracle Utilities Customer To Meter uses the following types of shipping memos:

- **General**: the shipping memo category that should be used when manually creating shipping memos

Oracle Utilities Work and Asset Management and Oracle Utilities Operational Device Management include additional types of shipping memos.

## Maintaining Shipping Memos

You use the **Shipping Memo** portal is used to display and maintain shipping memos.

Refer to About Shipping Memos for additional functional information.

You can access the portal from **Menu > Inventory > Shipping Memo**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.



Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Shipping Memo**: This zone provides information about the shipping memo, such as shipping category, shipping memo number, ship to details, and tracking information.
- **Shipping Lines**: This zone provides links to the shipping line and stock items associated with the shipping memo.

# Specifications


- About Specifications
- Maintaining Specifications


## About Specifications

Specifications are used to define the manufacturer, model, and other information about a set of assets.

- Specifications can include attachments which support the specification definition or provide the actual specification from the manufacturer.
- Specifications may also include default in service and out of service maintenance schedules for assets or components linked to the specification. When the specification is associated with the asset or component, the related maintenance schedule becomes in effect for the asset or component.

Specifications are defined by the following:

- **Asset Type**: The asset type associated with the specification. This links the specification to assets (assets of the specified asset type reference the current specification).
- **Manufacturer**: The manufacturer associated with the specification.
- **Detail**: Additional details pertaining to the specification. Details can include
  - **Model**: The model number for the specification. Used with item specifications.
  - **Configuration Mapping**: Details of the template device used when creating devices (meters, items, or communication components) based on assets that reference this specification. This includes:
    - **Configuration**: The type of configuration
    - **Device Identifier Type**: The device identifier type (Asset ID, Badge Number, Serial Number, etc.) used to identify the template device
    - **Identifier Value**: The value for the specified device identifier type used to identify the template device (for example, the Badge Number of the template device)
- **Firmware Information**: Details concerning the firmware specifications associated with the specification (not used with analog meter specifications)

The base package includes the following specifications, each based on a different business object. The table below lists the base package specifications and the business objects for each.

| Specification | Business Object |
|---|---|
| Analog Meter Specification | W2-ManualMeterSpecification |



Copyright © 2026, Oracle and/or its affiliates.

| Specification | Business Object |
|---|---|
| Item Specification | W1ItemSpecification |
| Communication Component Specification | W2-CommunicationComponentSpec |
| Communication Relay Specification | W2-CommunicationRelaySpec |
| Firmware Specification | W2-FirmwareSpecification |
| Analog Meter Specification | W2–ManualMeterSpecification |
| Smart Meter Specification | W2-SmartMeterSpecification |

## Maintaining Specifications

You use the **Specification** portal is used to display and maintain specifications.

Refer to About Specifications for additional functional information.

You can access the portal by selecting **Menu**, then **Asset Management**, then **Specification**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Specification**: This zone displays the basic attributes of the specification.
- **Specification Assets**: This zone displays a list of assets that reference the specification.

# Device Management

- Devices
- Device Configurations
- Measuring Components
- Measuring Component Comparison Periods

# Devices

- About Meters
- About Items
- About Communication Components
- Maintaining Devices

## About Meters


Copyright © 2026, Oracle and/or its affiliates.

Meters are a type of device, which can be physical or virtual objects, that can produce data to be handled by the system.

A meter can have one or more device configurations over time. A meter's measuring components (if applicable) are associated with the device via device configurations.

A meter's device category (defined as an option on the device's business object) defines the broad category to which a particular device belongs. This option is used by smart meter commands and other processes when determining how the system should behave when executing processes based on the category of device. For example, when viewing items via the Device portal, certain item-specific zones appear that are not displayed when viewing meters or communication components. The base package supports the following device categories for meters:

- **Smart Meter**: these meters are associated to a head end system and support smart meter commands. The measuring components associated to these devices are expected to be read very frequently. These meters support periodic estimation.
  Refer to About Periodic Estimation for additional functional information about how the periodic estimation can fill in missing reads for both interval and scalar measuring components.
- **AMR Meter**: these meters may also be associated to a head end system but they do not support smart meter commands. The measuring components are read frequently.
- **Manual Meter**: these meters are seldom read and do not support smart meter commands.

Attributes used to define devices include the following:

- Basic information about the Device including type of device (Smart Meter, Manual Meter, and so on), Manufacturer, and Model
- One or more device identifiers such as serial number, badge number, meter number, etc.
- For smart meters the device can also specify a head end system and an incoming data shift. Note: these fields are only necessary when there is no fallback defined on the device type.

Refer to Initial Measurement Data for additional functional information about how the device's head end system and incoming data shift can impact processing.

### About Communication Modules

Communication modules enable automatic meter reading for manual meters, and handle electronic transmission of measurement data and events for meters that are not capable or not enabled to handle this data. Devices of this sort are attached to manual meters as equipment, and in the case of modules that measure consumption or other data, are associated to the physical meter’s scalar measuring components.

Communication modules associated with a manual meter are defined in the **Device Equipment** list on the Manual Meter business object (D1-ManualMeter). See the embedded help for more information.

### Creating Measuring Components for Communication Modules


Copyright © 2026, Oracle and/or its affiliates.

Measuring components used with communication modules must be scalar register measuring components (based on the D1-Register business object) and must be defined on the manual meter's device configuration.

If the communication modules will also record interval consumption, create an interval channel measuring component (based on the D1-IntervalChannel business object) add it to the manual meter's device configuration, and add it as the register measuring component's **Consumption Reference Measuring Component**.

**Measurement and Device Event Processing for Communication Modules**

The IMD Seeder and Device Event Seeder support payloads that reference the communication module identifiers currently attached to a physical meter, instead of the physical meter's identifier.

Based on the physical meter and the communication module currently attached to it, the seeders can identify the physical meter's register (or related interval) measuring component and use it for instantiating the initial measurement or device event.

**Devices and Characteristics**

Characteristics can be defined for devices using the **Device Characteristics** zone on the **Device** portal. Only characteristics of types defined for the device's type in the **Valid Device Characteristic Types** zone on the Device Type portal can be defined for the device.

# About Items

While the most common type of devices are meters, there are many other devices that can be involved with a customer's service. These other devices are referred to simply as "items", and are used to represent lamps, poles, current transformers, backflow devices, pulse initiators, etc.

Items can be either "badged" or "unbadged". "Badged" items are those that have unique identifiers, and are represented by device records. Device records are not created for "unbadged" items. "Badged" items are installed at service points in much the same way as meters are installed (though they use an item-specific install event). For "unbadged" items, "multi-item" service points are used to define the number of each type of item that is installed at the service point.

These "badged" items are not directly measured and therefore do not have a measuring component. Instead there is configuration on the device type that will identify an effective dated average daily usage. Using the average daily usage the item's consumption will be calculated. The total consumption for a given period will always be equal to the average daily usage multiplied by the duration of the period. However, when the consumption for a period is turned into an interval curve there are two ways this can be done:

- **Use Straight Line**: the consumption will be evenly divided amongst the intervals (thus producing a straight line when graphed)


Copyright © 2026, Oracle and/or its affiliates.

- **Use Profile**: a profile measuring component will be used to align the consumption to a curve. This is useful because it can more accurately identify the hours where consumption has occurred.

Refer to About Service Points for additional functional information about "unbadged" items.

# About Communication Components

Communication components are devices attached to meters. A communication component may handle electronic transmission of measurement data, events, and commands in situations in which the underlying meter is not capable or not enabled to handle this data. Devices of this sort are sometimes referred to as ERT (Electronic Receiver/Transmitter) meters, or communication modules (for example, the term "gas module" may refer to a communication module attached to a gas meter).

# Maintaining Devices

You use the **Device** portal to display and maintain devices.

Refer to About Meters, About Items, and About Communication Components for additional information about specific types of devices.

You can access the portal by selecting **Menu**, then **Device**, then **Device**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Device** portal includes the following zones:

- **Device**: defines the basic attributes of the device
- **Device Configuration List**: lists the device configurations for the device
- **Device Attachments**: lists attachments related to the device
- **Device Activities**: lists activities related to the device in date descending order
- **Device Exceptions**: lists the first 50 VEE exceptions for all measuring components for the device, sorted by the exception creation date in descending order
- **Device Events**: lists the most recent device events associated to the device in descending order
- **Device Characteristics**: displays characteristics defined for the device. To add device characteristics, click the **Edit** link in the zone title bar.

> **Note**
>
> Only characteristics of types defined for the device's type in the **Valid Device Characteristic Types** zone on the **Device Type** portal can be defined for the device.

- **Device Entity Groups**: This zone lists the Entity Groups to assigned to the device. To add an entity group to the device, click **Add Entity Group to Device**. A dialog opens where you can select the entity group.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When asset management functionality is installed with Oracle Utilities Meter Solution Cloud Service, devices are maintained using the **Device** tab of the **Asset** portal (see Maintaining Assets).

# Device Configurations

- About Device Configurations
- Maintaining Device Configurations

## About Device Configurations

Device configurations represent specific configurations of a device as of a certain time.

Over time, a device can have many configurations. Device configurations are effective-dated allowing a device to retain its identifiers (such as serial number, badge number, etc.) even while the quantities it measures change over time.

Each device configuration can have one or more measuring components associated with it.

Attributes used to define device configurations include the following:

- The device configuration type
- The parent device
- The effective date and time of the device configuration
- The time zone in which the device operates

Refer to Multi Time Zone Support for additional information on the importance of the Device Configuration time zone.

## Maintaining Device Configurations

You use the Device Configuration portal to maintain device configurations.

Refer to About Device Configuration for additional functional information.

You can access the portal from **Menu**, then **Device**, then, **Device Configuration**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Device Configuration**: Defines the basic attributes of the device configuration
- **Measuring Components**: Lists the measuring components for the device configuration
- **Scalar Readings**: Displays a list of scalar readings for scalar measuring components associated with the current device configuration. To create a new reading for scalar measuring components associated with the device configuration, click **New Reading**. To edit an existing reading, click the edit icon for the reading you wish to edit.

**NOTE**: The Scalar Readings zone is available only with Oracle Utilities Meter Data Management.


Copyright © 2026, Oracle and/or its affiliates.

# Measuring Components


- About Measuring Components
- Physical Measuring Components
- Scratchpad Measuring Components
- External Measuring Components
- Maintaining Measuring Components
- Uploading Measurement Data
- Requesting Measurement Data


## About Measuring Components

Measuring components are single points for which data will be received and stored in the system.

A measuring component can be associated to a device, which can have one or more measuring components, or it can be "virtual" or "stand-alone," meaning that it is not associated to a device.

The base package supports the following types of measuring components:

- **Physical**: Physical measuring components are those that physically exist, and that are linked to a device that can be configured differently over time. Interval channels and scalar registers are examples of physical measuring components.
  Note: The terms register and channel are synonyms for measuring component.
- **Standalone**: A standalone measuring component is used to record measurements for something that does not have a physical presence or is not associated to a particular device or service point. For example, you might create a standalone measuring component to record the average daily temperature supplied by a weather station. Standalone measuring components can be identified through **Standalone Identifier**.
- **Scratchpad**: Users create scratchpad measuring components based on physical measuring components to experiment with measurement manipulation functions.
- **External**: External measuring components are used to model measuring components for data that is not stored in the system. For example, some meters may have channels (measuring components) that capture Amps or other data not used my Meter Data Management or Customer to Meter, but that should be passed on to other external systems such as Oracle Utilities Data Intelligence. Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring components that don't exist.
- **Aggregator**: An aggregator measuring component holds summarized usage from other measuring components. For example, aggregator measuring components could be configured to hold total consumption for each postal code within a service territory. Head end system processing statistics used by Oracle Utilities Smart Grid Gateway are stored as aggregated measurements for aggregator measuring components.
  Refer to About Aggregations for additional functional information about aggregation measuring components.

Attributes used to define measuring components can include the following:



Copyright © 2026, Oracle and/or its affiliates.

- The measuring component type
- The device configuration to which the measuring component is associated
- Details concerning how the measuring component is read, such as the number of digits, the type of read out (dials or digital), the meter multiplier, etc.
- **Initial Load IMD Ordering**: A flag that indicates if incoming initial measurements for the measuring component should be processed using Measuring Component Ordering. This is used most often in the case of daily scalar measuring components for which data is occasionally sent to the system out of order, but can also be used to ensure that "check channels" are processed before related channels. Applicable only to measuring component types used with Direct Measurement Processing (see Direct Measurement Processing and Measuring Component Ordering for more information).
- The VEE groups used for validation and estimation of measurement data. Refer to About VEE for details about the types of VEE groups supported.
- **Enable ML Based Validation**: A flag that indicate (Yes or No) if initial measurements for the measuring component will be processed through a machine learning scoring model. IF set to "Yes", VEE processing for initial load data will only be used when data fails machine learning normalcy checks. See Machine Learning Anomaly Scoring for more information.
- **Anomaly Score Threshold**: The anomaly score threshold for the measuring components. This overrides the value set for the measuring component's type and/or in the MDM Master Configuration. This field is only visible when the **Enable ML based Validation** flag is set to "Yes" or **Enable ML Based Validation (Fallback)** flag on the measuring component type is set to "Yes". See Machine Learning Anomaly Scoring for more information.

### Scalar vs. Interval

Beyond the four types described above, measuring components generally fall into one of two primary classes of: scalar measuring components, and interval measuring components.

- **Scalar measuring components** are measured at unpredictable intervals. For example, "once a month" is not a predictable interval as the amount of time between reads in unpredictable and inconsistent.
- **Interval measuring components** are measured at predictable intervals, such as every 15 minutes, every 30 minutes, every hour, etc.
- The terms "interval size" and "seconds-per-interval" (SPI) are used to define the size of an interval measuring component's intervals.
- Note: A device may have any combination of interval and/or scalar measuring components

### Measuring Component Measurements

Measuring components are configured to measure specific types of quantities. These include:

- **Unit of Measure:** The unit of measure for the quantity being recorded. Examples include kilo-watt hours (kWh), kilo-watts (kW), therms, cubic feet (CCF), temperature (Fahrenheit or Celsius), etc.
- **Time of Use:** Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.
- **Service Quantity Identifiers:** Used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Generally, SQI is only used when multiple measuring components measure the same thing, but in different ways. A meter


Copyright © 2026, Oracle and/or its affiliates.

that measures both generation KWH and consumption KWH could use SQIs to differentiate between the two.

The combination of UOM, TOU and SQI define what a measuring component measures. TOU and SQI are optional, but UOM must be defined for all measuring components. For example, consider a meter (as illustrated in the image below) with two measuring components, both measuring the same unit of measure (kWh), but each measuring component measures consumption in different time of use (TOU) periods (peak and off-peak).

**Subtractive vs. Consumptive Measurements**

Another attribute that defines how measuring components measure quantities is the distinction between subtractive and consumptive measuring components.

A subtractive measuring component's usage is equal to the current reading (also known as the Stop or End Measurement or Reading) minus the previous reading (also known as the Start Measurement or Reading). To put this more simply:

Usage = End Reading - Start Reading

Most residential scalar meters are subtractive and in some cases interval meters can be subtractive as well . The table below lists a series of measurements for a subtractive measuring component.

| Date | kWh (Reading) | Usage |
|---|---|---|
| 01/01/2010 | 0 | |
| 01/31/2010 | 1000 | 1000 kWh |
| 03/02/2010 | 1780 | 780 kWh |
| 04/01/2010 | 2700 | 920 kWh |

A consumptive measuring component's usage is equal to its current measurement. Most residential interval meters are subtractive. For scalar, consumptive measuring components are often used to measure demand, such as KW. The table below lists a series of measurements for a consumptive scalar measuring component.

| Date | KW (Reading) | Usage |
|---|---|---|
| 01/01/2010 | 0 | 0 KW |
| 01/31/2010 | 10 | 10 KW |
| 03/02/2010 | 10 | 10 KW |
| 04/01/2010 | 13 | 13 KW |

# Physical Measuring Components

Physical measuring components are associated to the meters installed at service points. Through a device configuration there can be one-to-many measuring components in place at a given point in time. Much like other measuring components these are either scalar or interval and either consumptive or subtractive.

> **Note**
>
> Refer to About Measuring Components for additional information about scalar versus interval and consumptive versus subtractive measuring components.


Copyright © 2026, Oracle and/or its affiliates.

The base product supports a core set of physical measuring components, as described in the following sections.

### Interval Channel

These measuring components represent consumptive data that is recorded on a regular and consistent frequency as defined by the interval size. By definition interval data must fall on a specific interval boundary. For example, a meter that receives data once per hour cannot have a measurement at 1:30. Rather, the data must fall on either 1:00 or 2:00 not in between.

Data for interval channels is expected to be contiguous. However, in situations where it is not, there is no impact to the calculation of those intervals subsequent to the missing intervals. This is because the calculation of each interval is independent to all other intervals.

If there is a gap in the intervals it can be estimated through the periodic estimation process.

### Interval Channel - Subtractive

These measuring components have behavior characteristics of both interval channels and registers. Like the interval channel they receive data on a regular and consistent frequency that is expected to be contiguous. Like a register they use a subtractive consumption calculation and provide rollover validation.

Unlike the interval channel when there are missing intervals there is an impact on the calculation of those intervals subsequent to the missing intervals. This is because the calculation of a given interval is always computed by subtracting its reading from the last known prior reading. The following table describes an example of that scenario:

| Date/Time | Reading Condition | Reading | Measurement Condition | Usage |
|---|---|---|---|---|
| 01/01/2010 12:00AM | Regular | 0 | Regular | 0 kWh |
| 01/01/2010 01:00AM | Regular | 10 | Regular | 10 kWh |
| 01/01/2010 02:00AM | Regular | 20 | Regular | 10 kWh |
| 01/01/2010 03:00AM | Regular | 30 | Regular | 10 kWh |
| Missing measurements | | | | |
| 1/01/2010 12:00PM | Regular | 120 | Combined Quantity | 90 kWh |

Notice that the consumption for the 01/01/2010 12:00PM interval has become overloaded and covers the entire period for which no intervals have had been received. Also take note that the measurement condition is assigned a unique condition of Combined Quantity to signify that the consumption covers more than one interval.

A similar situation can arise when there has been an underestimation of consumption for a particular meter:

| Date/Time | Reading Condition | Reading | Measurement Condition | Usage |
|---|---|---|---|---|
| 01/01/2010 12:00AM | Regular | 0 | Regular | 0 kWh |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | |
|---|---|---|---|---|
| 01/01/2010 01:00AM | System Estimate | 10 | System Estimate | 10 kWh |
| 01/01/2010 02:00AM | System Estimate | 13 | System Estimate | 3 kWh |
| 01/01/2010 03:00AM | System Estimate | 16 | System Estimate | 3 kWh |
| 1/01/2010 04:00AM | Regular | 40 | Estimated | 24 kWh |

To help avoid these situations where an interval becomes overloaded with usage due to missing or estimated intervals the subtractive interval channel has the ability to initiate the calculation of missing intervals or re-calculation of estimated intervals when processing newly received intervals.

> **Note**
>
> Refer to Understanding Measuring Component Types link for additional information about how to configure subtractive interval measuring component types to perform prior and subsequent estimation.

**Register**

A register, often referred to as scalar, measuring component is used to represent infrequently received data from a meter. Typically these types of measuring components will only be read once per month aligning to the billing period for the customer.

The physical meter these measuring components represent often have dial readings where there are a finite number of digits that can register the consumption. When the maximum amount has been measured the meter will rollover and start counting once more from zero. To appropriately handle these scenarios this type of measuring component can be defined with a rollover tolerance that allows the system to accurately identify an invalid reading from a standard rollover. For example when a meter has 6 dials (a maximum of 999,999):

- Going from a reading of 999,990 to 20 would likely be valid and represent consumption of 30.
- Whereas going from a reading of 999,990 to 800,000 would be an invalid reading since it would represent consumption of 810,000.

Given the infrequent nature of these measuring components they are not eligible for the periodic estimation process.

**Register - Auto-Read**

An automatically read register processes nearly identical to a register. The key difference between these two types of measuring components is that the automatically read variant is expected to be read on a frequency of at least once per day.

These measuring components are eligible for periodic estimation since the information about their reading schedule allows for missing readings to be detected and estimated.

Given the frequency these meters are read and the nature of subtractive consumption calculation they can be configured to re-estimate any past or future estimates when actual data has been received. The actual readings allow these estimates to be refined since the total


Copyright © 2026, Oracle and/or its affiliates.

consumption for the estimated period can be calculated (current reading - last reading prior to estimation).

> **Note**
>
> Refer to Periodic Estimation for additional functional information about how the periodic estimation can fill in missing reads for both interval and scalar measuring components.

## Scratchpad Measuring Components

Scratchpad measuring components provide users with a means to manipulate "scratchpad" measurement data without affecting existing "live" measurement data. Examples of measurement manipulation might include experimenting with the impact of executing the "spike smooth" function on an initial measurement, or adding or removing intervals to or from the measurement.

Scratchpad measuring components are defined by the same attributes as "regular" measuring components, with the following additional attributes:

- **User:** The name of the user who created the scratchpad. All scratchpad measuring components "belong" to a user.
- **Source Measuring Component:** The measuring component upon which the scratchpad is based.

Like "regular" measuring components, scratchpad measuring components can be related to a device through a device configuration, but scratchpad measuring components are not considered "active" for purposes of install events, and are not used in usage calculations for usage subscriptions.

Like "regular" measuring components, scratchpad measuring components and viewed and maintained using the Measuring Component portal, and measurement data for scratchpad measuring components can be viewed using the Initial Measurement portal or the 360 Degree View.

## External Measuring Components

External measuring components are used to model measuring components for data that is not stored or used in the system. For example, some meters may have channels (measuring components) that capture data not used my Meter Data Management or Customer to Meter, but that should be passed on to other external systems such as Oracle Utilities Data Intelligence.

For example, if your implementation has interval meters that record 2 channels of data (kWh and Amps), but only the kWh channel is used for usage calculations, you would set up the kWh channel as a Physical interval channel and the Amps channel as an External channel. Data for the kWh channel would be processed as usual, but the data for the Amps could either be filtered entirely, or passed through to an external system.

Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring components that don't exist. See **Filtering Data for External Measuring Components** below for more information.


Copyright © 2026, Oracle and/or its affiliates.

External measuring components are defined by the following;

- Device Configuration
- Channel ID
- External ID
- MDM External ID
- Number of Digits Left (Used for consumption calculations when passing data through to an external system)
- Number of Digits Right (Used for consumption calculations when passing data through to an external system)
- Channel Multiplier (Used for consumption calculations when passing data through to an external system)

The base product supports the following external measuring components:

- Interval Channel - External (D1-IntervalChannelExternal)
- Scalar Channel - External (D1-ScalarChannelExternal)

Like "regular" measuring components, external measuring components are related to a device through a device configuration, and can be considered "active" for purposes of install events, but are not used in usage calculations for usage subscriptions.

Like "regular" measuring components, external measuring components and viewed and maintained using the Measuring Component portal.

#### Filtering Data for External Measuring Components

Incoming data for external measuring components can be filtered (deleted) or passed through to an external system via object storage. Note that payload statistics track the number of initial measurements filtered and passed through.

Filtering data for external measuring components is supported with Direct Measurement Processing ONLY. See Direct Measurement Processing and Measuring Component Ordering for more information.

## Maintaining Measuring Components

You use the **Measuring Component** portal to maintain measuring components.

Refer to About Measuring Components for additional functional information.

You can access the portal by select **Menu**, then **Device**, then **Measuring Component**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Measuring Component**: defines the basic attributes of the measuring component
- **Initial Measurement Data History:** lists the last 50 initial measurements for the measuring component in reverse chronological order.
- **Measuring Component Profile Use:** lists all profile factors on which the measuring component is referenced as a factor value.


Copyright © 2026, Oracle and/or its affiliates.

- **Measurements**: lists final measurements for the measuring component, including any derived values for each measurement. Note: depending on the type of measuring component this zone may be the Interval, Scalar, or Subtractive Interval variant.
- **Device Configuration Overview**: Displays an overview of the measuring component's parent device's configurations over time, including the measuring components associated with each configuration, and service points (along with the relevant install events) at which the device's configurations have been installed over time, if any exist.
- **Data Requests**: Lists data requests for the measuring component. Applicable only for Standalone Data Request measuring components (based on the Standalone Data Request MC business object)
- **Measuring Component Comparison Period Details**: Lists comparison periods for the measuring component (created when the current measuring component is compared to another measuring component via the Interval Proxy Day Estimation VEE rule)

# Uploading Measurement Data

Measurement data for a measuring component can be manually uploaded in comma separated values (CSV) format. This can used to import data from external sources, such as weather systems or other market participants such as system operators in a settlement market.

This feature is accessed via the **Load IMDs (CSV)** option on the measuring component's context menu. When this option is selected, the **Online IMD CSV Upload** screen opens. From this screen users can select a CSV file containing measurement data to upload from a local desktop.

Import files can contain one or more initial measurements For interval data, breaks between initial measurements will be based on a gap greater than one interval size (SPI) between intervals. For scalar measurements, each row will create a separate initial measurement.

Refer to the embedded help on this page for details concerning the format of CSV files used for importing measurement data.

When processed, each initial measurement in the import file results in the creating of an initial measurement for the measuring component.

# Requesting Measurement Data

The base package provides a mechanism by which standalone measuring components can request data from an external application. Examples of this include requesting data weather data from a weather station or zonal load data from an independent system operator (ISO) when used with Oracle Utilities Market Settlements Management. Measuring components of this type are referred to as "standalone measuring components".

### Standalone Data Request Measuring Components

Requesting measurement data for standalone measuring components is supported by the "Standalone Data Request MC" (D1-StandaloneDataRequestMC) and "Standalone Data Request MC Type (D1-StandaloneDataRequestMCType) business objects.

Standalone data request measuring components use many of the same attributes as device-based measuring components, but also use the following unique parameters:

- **Standalone Identifier**: An identifier for the measuring component. This ID is used when importing data for standalone measuring components via the standard initial measurement loading process (see About Initial Measurement Data for more information).


Copyright © 2026, Oracle and/or its affiliates.

- **Processing Timetable Type**: A processing timetable type used when scheduling data requests. If not specified, a fallback processing timetable type can be defined on the measuring component's type.
- **Service Provider**: The external application to which data requests are sent. If not specified, a fallback external application can be defined on the measuring component's type.
- **Request Information**: Details about the last data request processed for the measuring component

### Measurement Data Requests

Requests for measurement data can be scheduled or made manually by a user.

- Scheduled requests are triggered by the "Standalone MC Data Request - Monitor" batch process.
- Manual requests are triggered by clicking the **Request Data** button in the **Record Actions** section of the **Measuring Component** zone.

  This opens the Request dialog where te user enters a start and end date/time for the request.

Each of these invokes the "Calculate Schedule and Create Outbound Communication" algorithm which creates an outbound communication based on the processing method defined for the external application (Service Provider). When the external application sends back the requested measurement data, it can be received via the standard method for importing measurement data.

### Configuring External Applications for Measurement Data Requests

To support measurement data requests, the following processing method must be defined for the external applications to which data requests are sent:

| Processing Role | Processing Method |
|---|---|
| Standalone MC Data Request | How to Send MC Related Information<br>Used to send the outbound communication to the external application.<br>**Default Communication Out Business Object**: D1-StandaloneMCDataRequest (Standalone MC Data Request Communication Out<br>The **Request Mapping Algorithm** and **Response Mapping Algorithm** options can be used to map the format of data both from the standard format to an external system format and from an external system format to the standard format.<br>For example, when requesting for zonal load from an ISO the process would be:<br>• Take the input of an outbound communication that has the measuring component's standalone ID and the date/times and put it into the zonal request format for the ISO<br>• Take the input of an outbound message that has the response from the ISO with the intervals and convert it into the IMD format |

In addition, the following objects must be configured to send data requests.:

- Communication Type for "Standalone MC Data Request Communication Type"
- External System
- Outbound Message Type


Copyright © 2026, Oracle and/or its affiliates.

- Message Sender

# Measuring Component Comparison Periods


- About Measuring Component Comparison Periods
- Maintaining Measuring Component Comparison Periods


## About Measuring Component Comparison Periods

Measuring component comparison periods are records of measuring component comparison. These are used by the Interval Proxy Day Estimation VEE rule to compare weather data to identify historical dates that most closely match current and/or future dates.

Measuring component comparison periods are created by the Interval Proxy Day Estimation rule, and are defined the following:

- **Primary Measuring Component**: The measuring component against which the comparison will be run
- **End Date** / **Start Date**: The date range that is compared to the period for the **Examined Measuring Component** (defined by the **Examined End Date** and **Examined Start Date**)
- **Rank**: The rank of this data point relative to the other comparison period stores for the same Measuring Component, Comparison Type, and End Date. The lower the rank, the better.
- **Examined Measuring Component**: The measuring component that will be compared to the **Primary Measuring Component**.
- **Examined End Date** / **Examined Start Date**: The date range that is compared to the period for the Primary Measuring Component (defined by the **End Date** and **Start Date**)
- **Lowest Quality Condition** : The lowest quality of condition code that was found for the measurements in the related period.
- **Comparison Metric**: The specific metric used to calculate the Comparison Result from the related Comparison Type)
- **Comparison Result**: The value result of the comparison.

## Maintaining Measuring Component Comparison Periods

You use the **Measuring Component Comparison Period** portal to maintain measuring component comparison periods.

Refer to About Measuring Component Comparison Periods for additional functional information.

You can access the portal from **Menu > Device > Measuring Component Comparison Period**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Comparison Period**: This zone displays the basic attributes of the measuring component comparison period.



Copyright © 2026, Oracle and/or its affiliates.

# Device Installations


- Service Points
- Service Point Quantities
- Install Events
- Route Management
- Facilities
- Network Locations

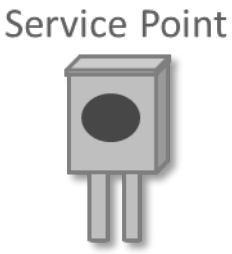

## Service Points


- About Service Points
- Maintaining Service Points


### About Service Points

Service points are physical locations (usually a meter socket) at which a company supplies service and are used in Oracle Utilities Meter Data Management to store information describing the type of service supplied and how it is measured. Devices are installed at service points, and the relationship between individual service points and devices can change over time. For example, over time different devices may be installed at a service point, and a device may be installed at different service points.

A service point may only have one device. If multiple devices at a service point are required, define a parent and child service point relationship to create a hierarchy of service points. On each service point, users will be able to specify a parent service point. An example of this feature would be to link each condominium in a building to a parent service point that represents the common areas of the building. Another example would be a deregulated service point that allows multiple devices per service point. In this example, the parent service point would be the market service point ID and the children service points would relate to the individual devices.




Attributes used to define Service Points can include the following:

- Basic information about the Service Point, including address, time zone, market, parent Service Point (if applicable), status, and main contact
- Specifics related to whether or not the current Service Point supplies service to life support equipment, or if the load supplied to the Service Point is considered sensitive



Copyright © 2026, Oracle and/or its affiliates.

- Information related to field work performed at the current Service Point, including any applicable warnings or instructions
- The measurement cycle, route, and route sequence for the Service Point

#### Service Point Access via Access Groups and Divisions

Access to service points (and objects and data related to the service point, see below) can be restricted based on Access Groups and Divisions.

#### Restriction via Access Groups

Service point data access restriction based on Access Group is enabled by default. Users can only access service points (and related objects and data) with an Access Group associated with the user via Access Role on the **Access Security** tab of the **User Preferences** portal. See The Big Picture of Row Security in the *Administrative User Guide* for more information about Access Groups and Access Roles. Restriction via Access Groups applies to all service point queries, including use of the **Service Point Search** zone, the **Route Management** and **360 Search** portals, and service point queries on other zones and portals.

#### Restriction via Division

To enable service point data access restriction based on Division, select "Yes" from the **Restrict Division** drop-down list in the **Control By Division** section of MDM Master Configuration.

When this restriction is in effect, users can only access service points (and related objects and data) with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. This applies to all service point queries, including use of the **Service Point Search** zone, the **Route Management** and **360 Search** portals, and service point queries on other zones and portals.

#### Unbadged Items

Items that do not have any unique identification are referred to as "unbadged." These types of items, unlike "badged" items, do not have a device to represent them. Instead they are associated to service points with the service point category of "multi-item." For these types of service points a list of "unbadged" items can be configured to represent the items that require usage to be calculated. This is an effective dated list that allows for changes over time to the type and count of items over time. The consumption of the list of items is either calculated using the service quantity associated to the item types listed or by using the installation override quantity that can be specified on the service point.

Refer to Multi Time Zone Support for additional information on the importance of the Service Point time zone.

## Maintaining Service Points

You use the **Service Point** portal to view and maintain service points.

Refer to About Service Points for additional functional information.

You can access the portal by selecting **Menu**, then **Device Installation**, then **Service Point**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When searching for services points or creating new service points, if the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal will be available. In addition, the **Division** drop-down list will only list divisions associated with the user.
> See **Restriction via Division** in About Service Points for more information.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point**: defines the basic attributes of the service point
- **Service Point Attachments**: lists attachments related to the current service point
- **Device History**: lists all device configurations that have been installed for the service point, in reverse chronological order
- **Usage Subscriptions**: lists all usage subscriptions linked to the service point, sorted in descending order by Start Date/Time of Usage Subscription/Service Point
- **Service Point Entity Groups**: This zone lists the Entity Groups to assigned to the service point. To add an entity group to the service point, click **Add Entity Group to Service Point**. A dialog opens where you can select the entity group.
- **Child Service Points**: lists the first 50 children service points of the current service point (displayed only if the current service point has child service points)

The following zones may appear as part of the portal's **Geographic Information** tab page:

- **SP/Measurement Cycle Schedule Route List**: lists all the measurement cycle schedule routes linked to the service point.
- **SP Network Locations**: lists all network locations linked to any facility associated with the service point, ordered by start date/time.

The following zones may appear as part of the portal's **Multi-Item** tab page:

- **Multi-Items**: defines "multi-items" installed at the service point. "Multi-items" are used to define configurations in which multiple unbadged items of a set of one or more item types are installed at the service point (for example, 3 items of one type and 4 items of another type).
- **Item Details**: displays details for a broadcasted "multi-item" in the **Multi-Items** zone. This zone only appears if a "multi-item" is defined in the **Multi-Items** zone.

The following zones may appear as part of the portal's **SP Activity** tab page:

- **SP Activity History**: shows all activities linked to the service point. The results can be filtered based on activity type category and creation date/time.
- **SP Service Issue Monitors**: displays a list of service issue monitors and related service investigative orders related to the current service point, in reverse chronological order.
- **SP Business Flags**: displays a list of business flags related to the service point in reverse chronological order.
- **SP Suppression History**: displays a list of suppression activities related to the service point in reverse chronological order.

The following zones may appear as part of the portal's **Market Settlement** tab page:


Copyright © 2026, Oracle and/or its affiliates.

- **SP Market Participants**: lists all market participants linked to the current service point
- **SP Quantities**: lists service point quantities for the current service point

The following zones may appear as part of the portal's **Log** tab page:

- **Service Point Log**: This zone displays the server logs for the service point.

# Service Point Quantities


- About Service Point Quantities
- Maintaining Service Point Quantities


## About Service Point Quantities

Service point quantities are quantities that are stored at the service point level. These quantities are calculated at irregular frequencies (such as annually or quarterly) and thus do not make sense as a measured quantity on a device and measuring component. For example, in some deregulated markets the Peak Load Contribution could be stored here and effective dated over time to represent the changing contribution of a given service point to the market wide coincidence peaks.

Service point quantities are defined by the following:

- **Service Point**: The service point for which the quantities were calculated
- **Service Point Quantity Type**: The Service Point Quantity Type used for the record
- **Start Date** / **End Date**: The date range for the calculated quantity
- **Service Point Quantities**: Calculated quantities based on **Quantity Identifiers** configured on the Service Point Quantity Type (see Understanding Service Point Quantity Types for more information)

> **Note**
>
> There cannot be more than one Service Point Quantity for the same Service Point, Service Point Quantity Type, for the same time period (defined by the Start Date and End Date parameters).

## Maintaining Service Point Quantities

You use the **Service Point Quantity** portal to view and maintain service point quantities.

Refer to About Service Point Quantities for additional functional information.

You can access the portal from the **Main > Device Installation > Service Point Quantity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point Quantity**: This zone displays the basic attributes of the service point quantity



Copyright © 2026, Oracle and/or its affiliates.

# Install Events


- About Install Events
- Maintaining Install Events


## About Install Events

Whenever a device is installed at a Service Point, an installation event is created. Installation events capture the history of the devices that have been installed at a Service Point. This allows consumption for a Service Point to be calculated over time. In technical terms, installation events (or Install Events) link a specific Device Configuration to a Service Point.

There are specific Install Event business objects for different types of devices, including smart meters, items, manual meters, and communication components.

When creating Install Events for items, only "badged" items (those that are uniquely identified in some way) use item Install Events. "Unbadged" items do not use Install Events. Instead, the number of each type of item is specified in the "Multi-Item Detail" section of the Service Point.

While a device is installed at a Service Point, it may be turned off (and back on again). The installation event that records the original installation date and time also records the dates and times when the device has been turned on and off. When a device is removed, the original installation event is updated with the removal date and time.

## Maintaining Install Events

This portal is used to display and maintain an Install Event.

Refer to About Install Events for additional functional information.

You can access the portal from **Menu > Device Installation > Install Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Install Event**: defines the basic attributes of the Install Event, the on/off history for the Install Event, and removal information.

When creating a new record, the following options are available:

| Name | Details | Business Object |
|---|---|---|
| Communication Component Install Event | This Install Event is intended for use with a Communication Component that is installed on top of a traditional Meter. | D1-CommComponentInstallEvent |
| Item Installation Event | This Install Event records the installation of an Item (i.e. streetlight). | D1-ItemInstallEvent |
| Manual Meter Installation Event | This Install Event is used for manually read meters. | D1-ManualMeterInstallEvent |



Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Smart Meter Installation Event | This Install Event is unique to smart meters and has a lifecycle that's maintained by remote meter commands. | D1-SmartMeterInstallEvent |

# Route Management


- About Route Management
- Working with Route Management


## About Route Management

Route management involves managing the order of the Service Points within measurement cycles and routes.

When creating Service Points, each Service Point can reference a measurement cycle, route, and its sequence within the route. Over time, as Service Points are added and removed to and from measurement cycles and routes, it may become necessary to change the sequence of Service Points within the measurement cycles and routes used in your implementation.

- A **measurement cycle** defines WHEN the Service Point is visited
- A **route** within a cycle defines a group of Service Points in a cycle that are visited by a meter reader
- A **sequence** within a route defines the physical position of the Service Point within a route
- A **schedule** specifies the dates on which Service Points are visited.

Manually read Service Point reference a measurement cycle, route and sequence within the route. A batch process creates SP/Measurement Cycle Schedule Routes for each Service Point, which link the dates on a measurement cycle schedule to a measurement cycle route defined for the Service Point. These define the specific date on which a meter reader will visit Service Points associated with a specific measurement cycle route, and the sequence in which the Service Points on that route are visited.

### Measurement Cycles

Measurement cycles define the schedule for manual meter reading of devices at Service Points in that cycle. Measurement cycles can have one or more associated routes used to collect measurements.

Measurement cycles can also be configured to define when to create usage transactions for Usage Subscriptions associated to Service Points in the cycle.

### Measurement Cycle Schedules

Measurement cycle schedules define the dates on which devices are scheduled to be read for a given measurement cycle and the routes used to collect measurements for the measurement cycle.

## Working with Route Management



Copyright © 2026, Oracle and/or its affiliates.

This Route Management portal is used to change the sequence of Service Points within a measurement cycle and route and to transfer Service Points from one measurement cycle and route to another.

Refer to About Route Management for additional functional information.

You can access the portal from the **Main > Device Installation > Route Management**. You are brought to a query portal with options for searching. Once you enter your search criteria, a list of related Service Points is presented.

The following actions are available once a Measurement Cycle and Route are selected:

- **Renumber Sequence**: renumbers the sequence of selected Service Points for the route. Use the Renumber Sequence page to renumber the sequence of Service Points in a measurement cycle route.
- **Transfer Measurement Cycle Route**: Use this button to transfer the selected Service Points to another route. Use the Transfer Measurement Cycle Route page to transfer the selected Service Points to another measurement cycle route. You can also renumber the sequence of Service Points in their new route as part of the transfer.

# Facilities


- About Facilities
- Maintaining Facilities


## About Facilities

Facilities represent the network node level closest to the Service Point. In terms of electric networks, facilities typically represent transformers.

While "Facility" is the generic name for this object, many of the Business Objects created within Oracle Utilities Meter Data Management have a more specific name. One good example is the Transformer (D1-Transformer) Business Object.

## Maintaining Facilities

This portal is used to display and maintain a Facility.

Refer to About Facilities for additional functional information.

You can access the portal from **Menu > Device Installation > Facility**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Facility:** defines the basic attributes of the facility
- **Network Locations Related to Facility:** displays a list of network locations associated with the current facility in reverse chronological order.
- **Service Points Related to Facility:** displays a list of Service Points related to the current facility, in reverse chronological order.



Copyright © 2026, Oracle and/or its affiliates.

# Network Locations


- About Network Locations
- Maintaining Network Locations


## About Network Locations

Network locations define the location of a facility within a larger network.

Network locations hold the network nodes that provide service to a facility. In the case of electric networks, network locations define the feeder and substation associated with a transformer (facility). Because these network nodes can change over time, a facility can have many network locations (this is an effective-dated relationship). For example, for electric Service Points, whenever the substation or feeder that provides power to a given transformer is changed, a new record that references the latest substation and feeder should be created.

## Maintaining Network Locations

This portal is used to display and maintain a Network Location.

Refer to About Network Locations for additional functional information.

You can access the portal from **Menu > Device Installation > Network Location**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Network Location**: defines the basic attributes of the network location
- **Service Points Related to Network Location**: displays a list of Service Points related to the current network location, in reverse chronological order.

# Measurements


- Initial Measurement Data
- Final Measurements
- Interval Snapshot Mapping
- Reader Remarks
- Periodic Estimation
- Measurement Reprocessing
- Consumption Sync
- About Interval Prices and Price Sets
- Maintaining Interval Price Sets




Copyright © 2026, Oracle and/or its affiliates.

# Initial Measurement Data


- About Initial Measurement Data
- Legacy Measurement Processing
- Direct Measurement Processing
- Machine Learning Anomaly Scoring
- CSV Upload
- Importing Data From External Systems
- Using the IMD Control Staging Portal


## About Initial Measurement Data

Initial measurement data (IMD) is the term for measurement data in its initial (or raw) form.

Measurements read from a measuring component are referred to as "initial measurement data" (or initial measurements) and are used to record how much of the quantity (defined by UOM, TOU, and SQI) measured by the measuring component was consumed.

Initial measurement data for scalar measuring components contain a single "reading" or value, while initial measurement data for interval measuring components can contain multiple readings, one for each interval that falls between the start time and stop time of the measurement.

At a simple level, initial measurement data goes through the following process:

1. Initial measurements are loaded into the system.
2. Initial measurement data is validated, edited, and estimated (VEE). Please refer to About VEE for more details on the VEE process.
3. Initial measurements are converted into final measurements and stored in the Measurement (D1_MSRMT) table.
4. If using Oracle Utilities Meter Data Management, final measurements are used to calculate usage (bill determinants, etc.). Please refer to About Usage Calculation for more details on the usage calculation process.

There are two ways in which initial measurement data is processed:

- **Legacy Measurement Processing** involves creating records in the database for each initial measurement upon receipt. These records are updated through each stage of processing until the initial measurement is finalized, and are retained in the system until archived and/or deleted. Processing is defined based on the initial measurement data business object. Legacy Measurement Processing can be used with Initial Load, Manual, and Estimation initial measurement data. See Legacy Measurement Processing for more information.
- **Direct Measurement Processing** involves processing incoming measurement data in memory, based on algorithms defined on an IMD type. If using Measuring Component Ordering, this also creates staging records for each initial measurement upon receipt (staging records are deleted once the measurement has been processed). Direct Measurement Processing is only supported for Initial Load measurement data. It does not support Manual or Estimation initial measurements. See Direct Measurement Processing for more information.



Copyright © 2026, Oracle and/or its affiliates.

Only initial measurements can be edited directly by end users, and only when using Legacy Measurement Processing..

Quantities in an initial measurement can change during processing.

- "Pre VEE" quantities are consumption values derived from the measurements recorded by the head end system or meter reader with the appropriate installation and measuring component multipliers applied.
- "Post VEE" quantities are the "final" values, after VEE processing.

Changes in quantities can be based on a number of conditions, including:

- VEE rules have changed the quantities because they are missing or obviously wrong. In this, the Pre VEE values are adjusted based on the specifics of the VEE rules applied to the initial measurement to create the Post VEE values
- Manual changes by a user.

When using Legacy Measurement Processing, initial measurement data records contain both the original (Pre VEE) and final (Post VEE) versions of the quantities recorded by the measuring component.

When using Direct Measurement Processing, if any quantities are changed during VEE processing, original (Pre VEE) quantities are captured along with the final (Post VEE) quantities in the Measurement (D1_MSRMT) table.

In addition to recorded consumption values, measurements also have condition codes, used to indicate the source and quality of a measurement. For example:

- Regularly recorded measurements might have a condition code of "Regular"
- Missing measurements might have a condition code of "Missing"
- Estimated measurements might have a condition code of "External Estimated" or "System Estimated" based on where the estimation was performed.

Both Pre VEE and Post VEE values have their own condition code, which can also change during VEE processing. For example, a measurement that was missing in the Pre VEE document will likely become "System Estimated" in the Post VEE document once VEE has been executed.

# Legacy Measurement Processing

Initial measurement data (IMD) received from external systems is processed through two major stages:

- **IMD Seeder**: This is the common interface point that receives all IMD information and performs critical validations and conversions. During this process the IMD Seeder identifies the appropriate specific IMD business object for further processing. The choice is made based on the category of IMD being processed (typically initial load) and the type of measuring component the IMD is for. For example, an IMD for an interval measuring component would result in the Initial Load IMD (Interval) business object being selected. If for some reason the IMD Seeder encounters an error an instance of the IMD Seeder will be created to manage the resolution of that error.
- **Specific IMD**: Each specific IMD has the appropriate calculation and validation logic for the type of data that has been received. An IMD containing scalar data will be processed differently than interval data or subtractive interval data. However, all types of data will have a similar set of steps taken:


Copyright © 2026, Oracle and/or its affiliates.

- Any external system specific statuses for the measurement data are interpreted into the appropriate internal statuses
- An initial consumption calculation is performed
- VEE rules are executed
- Final measurements are created

**IMD Seeder Processing**

The IMD Seeder receives IMD data in a common structure, however, the content of the data (identifiers, date/times, etc) can vary depending on the system and the particular device that generated it. To accurately translate the external content to values understood by the system the IMD Seeder leverages the configuration of the service provider (aka head end or external system) and the device.

The service provider specifies the following inputs into IMD Seeder processing:

- **Identifier Types:** Each IMD must identify the device and measuring component the data has been generated from. The system supports a multitude of identifiers for both the device and measuring component. The service provider configuration specifies which identifier types are used for both the device and the measuring component which enables the IMD seeder to search for the correct destination for the measurement data. For example, a service provider might indicate that devices are identified by a serial number and measuring components are identified by a channel identifier.
- **Date/Time Format:** The service provider configuration will identify whether the date/times that were sent included specific time zone designations. This allows the system to accurately handle any time zone conversions that may be necessary.
- **Processing Methods:** A service provider has many processing methods that are used in the evaluation of the IMD.
  - Initial Measurement Creation - Maps a given measuring component type to the initial measurement BO that should be created
  - UOM Translation - Identifies the extendable lookup that should be used to map the service provider specific UOM to the internal UOM
  - TOU Translation - Identifies the extendable lookup that should be used to map the service provider specific TOU to the internal TOU
  - SQI Translation - Identifies the extendable lookup that should be used to map the service provider specific SQI to the internal SQI
  - Time Zone Translation - Identifies the extendable lookup to be used in mapping the service provider specific time zone to the interval time zone.

The device specifies the following inputs into IMD Seeder processing:

- **Incoming Data Shift:** This value is either specified directly on the device or can be defaulted via fallback on the device type. It identifies whether the incoming date/time information has is "Shifted" meaning it has been adjusted for Daylight Savings Time (DST) when in effect, or it is "Not Shifted" which means it will never be adjusted for DST. This plays an important role in the date/time processing since all incoming IMD date/times will be stored without an adjustment for DST.

With these inputs the IMD Seeder is able to translate the external identifiers into the appropriate identifiers to tie the IMD data to a specific measuring component in the system.

Once a measuring component is identified it is then used to select the appropriate IMD business object using the "Initial Measurement Creation" processing method of the service provider.


Copyright © 2026, Oracle and/or its affiliates.

### IMD Seeder Critical Validations

For Initial Measurement Data (IMD) to continue on to specific IMD processing it must pass several critical validations:

- The measuring component must be identified.
- The IMD must contain the appropriate date/times for the type of IMD being processed: scalar must have an end date/time and interval must have both a start and end date/time.
- The interval list must match the number of intervals expected based on the IMD start and end date/times. Note: if there are too few intervals the missing intervals can be automatically added to the IMD without generating an error. How this is done depends on whether there are date/times for each individual interval. If there are date/times for each interval the missing intervals are placed where the gaps exist. Otherwise when there are no date/times for the individual intervals the missing intervals are added to the end of the interval list.

If any of the critical validations have failed the IMD processing will be halted and an instance of the IMD Seeder will be created

Note: There is special logic that exists for handling the situation where either the IMD start date/time or the IMD end date/time falls on the duplicate hour that occurs when a time zone transitions out of daylight savings time. Typically there are two 1AMs on that day, one is considered to be during DST and the other is not. To ensure that the appropriate hour is selected the IMD Seeder uses the count of intervals and final measurements that exist prior to the IMD to identify the appropriate hour.

> **Note**
>
> Additional detail on the logic to derive the service provider, device, and measuring component identification can be found on the D1-DER-SPRMC Algorithm Type.

### IMD Seeder Time Zone Translation and Conversion

To provide the most accurate tracking of measurement data all initial measurement data is stored in a single time zone without an adjustment for Daylight Savings Time (DST). As utilities can have multiple head end systems that each send date/time information in different ways and some utilities operate in multiple time zones the IMD Seeder contains functionality to convert incoming date/times from a source time zone into the time zone of Oracle Utilities Meter Data Management (as indicated by the base time zone)

There are three components to identifying the source time zone of the incoming IMD:

- **Service Provider IMD Import Date/Time Format:** When the IMD date/times are sent in format that identifies the time zone for each date/time the date/times will be delivered to the IMD Seeder already converted.
- **Incoming Data Shift:** This indicates whether the date/times will be adjusted for DST while it is in effect. A value of "Always in Local Time" indicates that the date/times are adjusted for DST. A value of "Always in Standard Time" indicates that the date/times are not adjusted for DST
- **Time Zone**: The time zone of the incoming data is identified using the follow order of precedence:
  1. The time zone as indicated by the IMD data
  2. The service point time zone


Copyright © 2026, Oracle and/or its affiliates.

3. The device configuration time zone
4. The measuring component time zone
5. The base time zone

Once the time zone and the data shift of the incoming date/times is identified all date/times will be converted from that time zone and shift into the base time zone in standard time.

> **Note**
>
> Additional detail on the logic for date/time adjustments and validation of undercount and overcount can be found on the D1-DODTTMADJ Algorithm Type.

**Specific Initial Measurement Data Processing**

There are several categories of initial measurement data:

***Initial Load***

When initial measurement data is received from external systems (e.g. Head End systems) it is processed using the initial load IMD type. These IMDs are configured to execute the VEE rules that as associated to the Initial Load VEE role on the measuring component (or measuring component type in the case of a fallback). For adapter specific variations of the Initial Load IMD there is logic to map external measurement statuses to measurement conditions.

***Estimation***

For various reasons a meter will not report measurement data for a certain period of time. When the system detects that a measuring component is missing final measurements, it can create an initial measurement to execute estimation rules. This type of initial measurement is referred to as an estimated initial measurement.

At a high-level, the estimation process is as follows:

- Missing final measurements are detected by the Periodic Estimation process
- Estimated initial measurements are created for the "missing" time period by the Period Estimation process
- Values and consumption for the estimated initial measurements are calculated by "estimation" VEE rules.

It's important to note that the processes that detect missing measurements do NOT themselves estimate consumption. Rather, these detection processes simply create an initial measurement and let the estimation VEE rules estimate the consumption for the initial measurement.

Estimation IMDs execute the VEE rules associated to the Estimation VEE role on the measuring component (or measuring component type in the case of a fallback).

> **Note**
>
> Refer to Periodic Estimation for additional functional information about how the periodic estimation can fill in missing reads for both interval and scalar measuring components.

***Manual Override***


Copyright © 2026, Oracle and/or its affiliates.

Manual override IMDs are used when the creation of an IMD is user initiated as can be done from the various functions available on some of the 360 Degree zones (e.g., Final Values Overlay, Interval - Initial Measurement Overlay, etc.). Unlike other types of IMD manual override allow the user to edit the initial measurement data prior to performing VEE. These IMDs are used for more advanced tasks where a user must specify, either directly or through a measurement function, the values for the initial measurements.

These IMDs are systematically created during the reconciliation process that occurs when IMDs are finalized for measuring components with subtractive consumption calculations. They are used to recalculate the consumption on a measurement subsequent to an IMD when the start reading for that subsequent measurement has been changed.

Manual override IMDs execute the VEE rules associated to the Manual Override VEE role on the measuring component (or measuring component type in the case of a fallback).

***System***

A system is generated by internal system processes that need to recalculate measurement multipliers. They do not perform VEE but they can be configured to do so.

> **Note**
>
> Refer to About Measurement Reprocessing for additional functional information about how the system IMDs are created.

***Data Conversion***

This type of IMD has only the processing necessary to propagate converted measurement data into the system. The data provided to these IMDs is expected to be the quality of final measurements. Therefore no VEE rules will be executed.

**Scalar Initial Measurement Data Processing**

Initial measurement data for subtractive scalar measuring components contain start and stop readings in addition to Pre and Post VEE usage. For example, a set of initial measurements for a subtractive scalar measuring component might look like the following:

| Date/Time | Start Reading | Stop Reading | Pre VEE Usage | Pre VEE Condition | Post VEE Usage | Post VEE Condition |
|---|---|---|---|---|---|---|
| 01/01/2010 12:00AM | 0 | 1500 | 1500 | Regular | 1500 | Regular |
| 02/02/2010 4:11PM | 1500 | 2100 | 0 | Missing | 600 | System Estimate |
| 03/03/2010 5:22 PM | 2100 | 2900 | 800 | Regular | 800 | Regular |
| 04/01/2010 01:00PM | 2900 | 3500 | 0 | Missing | 600 | System Estimate |

Subtractive measuring components can "rollover" when the reading exceeds the maximum value based on the number of dials on the measuring component . For example, a register with a 4 dials can record values up to 9999 before rolling over to 0000. When this occurs, consumption is calculated based on the following attributes and calculated values.

- **Rollover Threshold** is the percentage of the measuring component's dial capacity at which measurements for measuring components of this type are considered to have rolled over. Dial capacity is the largest value that can be recorded for the measuring component,


Copyright © 2026, Oracle and/or its affiliates.

based on the measuring component's number of dials. For example, a measuring component with 5 dials has a dial capacity of 99999.

- **Number of Dials** is the number of dials used to record values on the register, and is equal to the **Number of Digits Left** field configured on the measuring component.
- **Max Dial Capacity** is the maximum value for the number of dials, rounded up to the next whole multiple of 10 (or 10 raised to the power of the number of dials). For example, for a register with 4 dials, the max dial value is 10000.
- **Max Acceptable Difference** is the maximum acceptable consumption that can be recorded for the register. This is equal to the Max Dial Capacity multiplied by the rollover threshold. For example, the Max Acceptable Difference for a register with 4 dials and a rollover threshold of 90% would be 9000. If the consumption is greater than this value, the initial measurement is transitioned to the Error state.
- **Difference**: The difference between the Stop Reading and Start Reading, obtained by subtracting the Start Reading from the Stop Reading. If the Difference is less than zero (<0), then add the Max Dial Capacity to calculate Rollover.
- **Rollover**: The adjusted consumption for a reading on a register that has rolled over. Only applicable if the Difference (Stop Reading - Start Reading) is less than zero (<0).
- **Consumption**: The calculated consumption for the reading, equal to either the Difference or Rollover. If the Difference is greater than or equal to zero, consumption is equal to the Difference. If the Difference is less than zero (<0), and the Rollover is less than or equal to the Max Acceptable Difference, the consumption is equal to the Rollover.

**Example**: Consider an initial measurement with the following attributes:

- Number of Dials: 4
- Rollover Threshold: 90 (%)
- Start Reading: 8900
- Stop Reading: 0500

For this reading,

- Max Dial Capacity = 10000
- Max Acceptable Difference = 9000 (10000 * 90)
- Difference = 0500 (Stop Reading) - 8900 (Start Reading) or **-8400**
- Rollover = 10000 (Max Dial Capacity) + -8400 (Difference) or **1600**
- Consumption is equal to **1600** (Rollover).

Each scalar IMD that is processed will perform a rollover validation based on the start reading. For those scalar IMDs that have measurements subsequent to the date being processed there will be an additional rollover validation forwards to ensure that the current IMD does not invalidate a future reading.

If the rollover validation fails when validating either backwards, forwards, or both then the system will attempt to automatically correct the situation by evaluating whether the measurements prior to or subsequent to the IMD can be removed. This analysis is done based on the condition of the measurement and the condition of the IMD. For example. If there is a rollover based on the start measurement and the start measurement is a system estimate but the IMD is regular then the start reading will be bypassed and the system will search backwards until:

- It identifies a measurement with a reading that does not cause a rollover when used as the start reading for the IMD


Copyright © 2026, Oracle and/or its affiliates.

- It encounters a measurement that causes a rollover but is also regular and therefore cannot be bypassed. At this point the IMD would be placed in an error status.

When a rollover error is identified that cannot be automatically corrected there are a few options available for fixing the situation:

- The "Override Condition" functionality can be used. This is an action button available on the IMD from within the Error status. It allows the user to provide a higher quality condition for the IMD to force the rollover process to bypass the measurement (either prior to or subsequent to the IMD) that was the source of the rollover threshold error. For example, if the start read was a regular measurement and the IMD was also a regular measurement the user could update the IMD to have a condition of "Super" which would allow the regular start reading to be bypassed.
- A Manual Override IMD could be created to adjust the measurements surrounding the IMD to ensure that the rollover error is no longer encountered.

> **Note**
>
> When a measurement is bypassed by the rollover automatic correction logic it will be set to Do Not Use when the IMD is finalized.

A scalar IMD can generate a manual override IMD during the finalization process when the need for reconciliation is identified. This happens whenever an IMD is received that occurs prior to an already existing scalar measurement. Since the newly processed IMD will introduce a new start reading for the existing subsequent measurement it is necessary to use the manual override IMD to recalculate the consumption. For example, imagine a scalar measuring component had the following readings:

| Date/Time | Start Reading | Stop Reading | Pre VEE Usage | Pre VEE Condition | Post VEE Usage | Post VEE Condition |
|---|---|---|---|---|---|---|
| 01/01/2010 12:00AM | 0 | 1500 | 1500 | Regular | 1500 | Regular |
| 02/02/2010 4:11PM | 1500 | 2100 | 0 | Regular | 600 | Regular |
| 04/01/2010 01:00PM | 2100 | 3500 | 1400 | Regular | 1400 | Regular |

When the February reading is finally received it will spawn a reconciliation Manual Override IMD to correct the consumption of the April reading given the new start reading:

| Date/Time | Start Reading | Stop Reading | Pre VEE Usage | Pre VEE Condition | Post VEE Usage | Post VEE Condition |
|---|---|---|---|---|---|---|
| 01/01/2010 12:00AM | 0 | 1500 | 1500 | Regular | 1500 | Regular |
| 02/02/2010 4:11PM | 1500 | 2100 | 600 | Regular | 600 | Regular |
| 03/03/2010 5:22 PM | 2100 | 2900 | 800 | Regular | 800 | Regular |
| 04/01/2010 01:00PM | 2900 | 3500 | 600 | Regular | 600 | Regular |


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> These reconciliation IMDs are generated during the finalization of the initiating IMD and must finalize themselves. If for some reason the reconciliation IMD does not complete it will prevent the initiating IMD from finalizing as well. Typically the reason for this issue will be a VEE exception that is being raised against the reconciliation IMD. The reason this rule is enforced is to ensure that the quantity of a measurement always aligns to the reading of that same measurement. If this situation arises the data source of the IMD can be used as an input to the VEE rule eligibility to skip certain rules when the IMD is a reconciliation IMD since the reconciliation process add a data source of "Generated by Reconciliation" to each reconciliation IMD.

### Interval Initial Measurement Data Processing

Given the consumption calculation for interval channels is done using the consumptive method the Interval IMD variants do not contain much special processing outside of the general pattern described in Processing Initial Measurement Data.

### Subtractive Interval Initial Measurement Data Processing

Initial measurement data for subtractive interval measuring components readings and usage for each interval in addition to Pre and Post VEE usage. Furthermore, there is a condition for both the reading and the usage which allows for more accurate recalculation of intervals. The start reading for each interval is considered to be the reading from the prior interval. For the first interval the start reading will be identified from the measurement data for the measurement immediately prior to the IMD. For example, an initial measurement for a subtractive interval measuring component might look like the following when we assume a start reading of 1490:

| Date/Time | Pre VEE Reading / Condition | Pre VEE Usage / Condition | Post VEE Reading / Condition | Post VEE Usage / Condition |
|---|---|---|---|---|
| 01/01/2010 01:00AM | 1500 / Regular | 10 / Regular | 1500 / Regular | 10 / Regular |
| 01/01/2010 2:00AM | 0 / Missing | 0 / Missing | 1520 / System Estimate | 20 / System Estimate |
| 01/01/2010 3:00AM | 1525 / Regular | 5 / System Estimate | 1525 / Regular | 5 / System Estimate |
| 01/01/2010 4:00AM | 1540 / Regular | 15 / Regular | 1540 / Regular | 15 / Regular |

Subtractive interval IMDs can validate for rollovers based on the configuration of the measuring component. See Scalar Initial Measurement Data Processing for more details on how rollovers are calculated.

How subtractive interval IMDs identify rollover errors is identical to scalar IMDs. However, the resolution of those errors is much different. With scalar any bypassed measurements are handled at the point of finalizing the IMD and the measurements are set to Do Not Use. With subtractive interval any bypassed measurements are added to the IMDs interval list so that they are processed along with the IMD that identified them as causing a rollover.

Subtractive interval IMDs also must initiate the reconciliation of an interval that comes directly after the IMD being processed. Unlike scalar this is not handled through a separate IMD. Rather the next measurement after the IMD being processed is added to the current IMD and the reconciliation of the usage for that measurement is computed as part of the IMD. This


Copyright © 2026, Oracle and/or its affiliates.

allows for any validation errors that are encountered in the recalculation of that measurement to be resolved along with the IMD.

> **Note**
>
> There is one scenario that does result in a reconciliation IMD being generated. When the last interval of an IMD has the reading changed during the VEE process it will require that a reconciliation IMD be generated to adjust the usage of the next measurement after the IMD (if applicable).

The subtractive interval IMDs are responsible for ensuring that the reading for each interval is in line with the usage calculated. This maintenance is performed each time the IMD is run through the VEE rules. After which the IMD will examine each interval and identify which intervals require an update to the reading or quantity.

#### Initial Measurement Data Valid Date Boundaries

Meter Data Management can be configured to define an acceptable date range for initial measurement processing, providing a mechanism to restrict either future measurements or very old measurements from being processed by Meter Data Management.

Defining this date range is done via the following parameters in the **Device Measurement Parameters** section of the MDM Master Configuration:

- **Prevent Future Measurement Records**: Indicates if the system should prevent processing for measurements deemed to be "future" measurements, based on the **Number of Days Before Considered Future** parameter. If a measurement is deemed to be a "future" measurement and this is set to "Yes", the initial measurement is moved to the "Error" state.
- **Number of Days Before Considered Future**: The number of days in the future (based on the IMD From Date Time) before a measurement is considered to be a "future" measurement. If the I**MD From Date Time** for a measurement is more days in the future than this setting, the measurement is considered "future".
- **Prevent Old Measurement Records**: Indicates if the system should prevent processing for measurements deemed to be "old" measurements, based on the **Number of Days Before Considered Old** parameter. If a measurement is deemed to be an "old" measurement and this is set to "Yes", the initial measurement is moved to the "Error" state.
- **Number of Days Before Considered Old**: The number of days in the past (based on the IMD From Date Time) before a measurement is considered to be an "old" measurement. If the **IMD From Date Time** for a measurement is more days in the past than this setting, the measurement is considered "old".

#### Deferring Initial Measurement Data Processing

Meter Data Management can be configured to defer processing for initial measurements that are in the past, but that are not yet deemed to be "old" (based on the **Number of Days Before Considered Old** parameter on the MDM Master Configuration). This ensure that a meter that "wakes" up and sends a large payload of data, including old measurements, will not adversely impact initial measurement processing during critical processing windows.

Configuring the system for deferring initial measurement processing is done via the following parameters in the **Device Measurement Parameters** section of the MDM Master Configuration:


Copyright © 2026, Oracle and/or its affiliates.

- **Defer Old Measurement Records**: Indicates if the system should defer processing of measurements whose **IMD From Date Time** falls between the **Number of Days Before Deferring** and the **Number of Days Before Considered Old** parameters. Note that when this parameter is set to "Yes", the **Number of Days Before Deferred** must be supplied.
- **Number of Days Before Deferred**: The number of days in the past (based on the **IMD From Date Time**) before a measurement should be deferred. Initial measurements with dates older than this setting but not as old as the **Number of Days Before Considered Old** will be deferred (the measurement is moved to the Defer state). If provided, this must be a positive number and must be less than **Number of Days Before Considered Old** setting.

Deferring initial measurement data processing requires that the Compare IMD Date/Times to Master Config Boundaries algorithm (D2-CMIMDDTMC) be defined as an Enter algorithm on the VEE Ready state for all initial measurement business objects for which deferring processing is to be enabled. See the **Detailed Description** of this algorithm for more information.

#### Initial Measurement Data Prioritization

By default, initial measurements are created and/or received into the system in an initial state of "Pending" and are transitioned to subsequent states via the "IMD Monitor - Physical Devices" Monitor process. This Monitor process can be skipped by setting the Execution Method flag on the initial measurement to "Real Time" (D1RT), in which case an Enter algorithm transitions the initial measurement to the next state in its lifecycle immediately. This feature can be used to prioritize processing of certain types of initial measurements.

The Execution Method flag can be set in initial measurements sent from the head end system, or can be inserted during Oracle Utilities Smart Grid Gateway Adapter processing.

Oracle Utilities Smart Grid Gateway Adapters set this flag to "Real Time" on all measurements created as the result of smart meter commands (Remote Connect, Remote Disconnect, or On-Demand Read) or completion events (including Field Activity Completion Events used with Service Order Management).

- Maintaining Measuring Component Dates

## Maintaining Measuring Component Dates

There are several date/times systematically maintained on measuring components that aid in the efficient processing of initial measurement data as well as informing the periodic estimation process.

During initial measurement data finalization these dates are maintained by a series of algorithms based on the type of measuring component and type of initial measurement.

#### Important Measuring Component Dates

Each initial measurement is evaluated to determine if the following dates should be updated on the measuring component.

#### Last Contiguous Read Date/Time

This represents the last point where an uninterrupted series of measurements were received for the measuring component. Periodic estimation uses this field to generate estimates for any periods of missing measurements for the measuring component.

For interval measuring components this date will be updated whenever the current value falls between the initial measurement's From and To Date/Time fields (inclusive) in one of two ways:


Copyright © 2026, Oracle and/or its affiliates.

1. If there are no measurements subsequent to the initial measurement being processed it will be set to the initial measurement To Date/Time
2. If there are measurements subsequent to the initial measurement and the initial measurement has filled a gap of missing measurements then it will be set to the Last Contiguous Measurement Date/Time that is subsequent to the initial measurement To Date/Time

For scalar measuring components there is an added check to ensure that the initial measurement's To Date/Time is exactly one expected measurement after the current last contiguous read date/time

**Adjusted Latest Measurement Date/Time**

For scalar meters this date is derived by assigning the last contiguous read date/time to an expected measurement date/time based on the measuring component type configuration for Expected Hours Between Measurements, First Daily Measurement Time, Early Threshold and Late Threshold.

For example, if a measurement is expected every 6 hours and the adjusted last contiguous read date/time on the measuring component is January 1st at 12:00:00AM, then the initial measurement's To Date/Time must align to the expected measurement date/time of January 1st at 06:00:00AM so the latest measurement date/time can be updated.

The expected measurement date/times are calculated using the expected hours between measurements and the first daily measurement time elements from the measuring component type. For example, if the expected hours between measurements was 6 hours and the first daily measurement date/time was 12:00:00AM, then the expected measurement date/times for each day would be:

- 12:00:00AM
- 06:00:00AM
- 12:00:00PM
- 06:00:00PM

To account for scalar reads that are not received for an exact date/time, the Early Threshold and Late Threshold elements of the measuring component type are used to align a given measurement to one of the expected measurement date/times.

**Most Recent Read Date/Time**

This represents the maximum measurement date/time that exists for the measuring component. This field is leveraged throughout the system to identify when certain logic must be executed. For example, if there are no measurements in the future to what is being processed then no logic dealing with future measurements need be executed.

It will be updated to the initial measurement's To Date/Time whenever the initial measurement's To Date/Time is greater than the current most recent read date/time.

**Most Recent Non Estimated Read Date/Time**

Represents the latest measurement processed that was not estimated. This is leveraged by the outage storm mode module.

It will be updated to the latest non-estimated measurement date/time in the initial measurement that is more recent than the current most recent non-estimated read date/time.


Copyright © 2026, Oracle and/or its affiliates.

### Identifying Data for Historical Extracts

Another important role for each of the algorithms that maintains measuring component dates is to identify whether the data being processed needs to be re-extracted for one of the existing measurement extract methods:

1. Specialized Measurement Extract
2. Specialized IMD Header Extract
3. DataConnect
4. Configurable Consumption Extract

If the data should be re-extracted a record will be created in the general processing table to be extracted at a later point.

It is important to note that by default these checks are turned off for performance reasons. If you would like to capture historical changes for any of these extracts it will need to be turned on for each individual algorithm.

### Date Maintenance Algorithms

The following algorithms are used to maintain measuring component dates.

**D1-UPD-DTMC - Update Latest Measurement Date/Time on MC with Consumption Sync**

This algorithm applies to initial load and manual initial measurements for interval measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

> **Note**
>
> If the first time Periodic Estimation will be executed after initial conversion the Latest Read Date/Time has not been initialized for any measuring components the "No of Hours in Past to Retrieve Last Useable Measurement" option of the Feature Configuration Type "Measurement Data Option" should be configured. This enables logic that will defer the initialization of the Last Read Date/time to the Periodic Estimation process for any measuring component where it is not currently populated but does have existing final measurements. Please refer to About Periodic Estimation for more information.

If the initial measurement's device configuration is configured for related measuring component consumption synchronization this algorithm can optionally identify whether there are any consumption synchronization eligible measurements on the related measuring component (eligible measurements are defined as those whose condition falls between the bottom and top ranges for consumption sync or the additional condition code for consumption sync). When the initiating initial measurement is not "back dated" or the synchronization period does not extend past the initial measurement's date range the appropriate interval or scalar consumption sync


Copyright © 2026, Oracle and/or its affiliates.

activity will be selected based on the measuring component type of the related measuring component. When the initiating initial measurement is "back dated" and there is a contiguous period of synchronization eligible measurements that extends past the date range of the initial measurement the Gap Consumption Sync Activity will be created.

Lastly this algorithm can be configured to maintain outage related activities for the device. Since the receipt of initial load data for a device indicates it is now powered on and communicating if there are any outstanding outage or estimation suppression activities (initiated via widespread outage) they can be ended as of the from date/time of the initial measurement.

**D1-UPD-DTMCF - Update Latest Date/Time on MC from a Future Measurement**

This algorithm applies to estimation initial measurements for interval measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

As this algorithm is used on estimation initial measurement data it is not common that it will update the most recent non-estimated read date/time. However, there are circumstances where high quality estimations based on a check channel might be given a measurement quality in the regular range. In our documentation we sometimes refer to these as prorated measurements.

It is also important to note that this algorithm will not initiate consumption sync processing.

**D1-UDTSCMCRE - Update Latest Measurement Date/Time on Scalar MC with Consumption Sync**

This algorithm applies to initial load and manual initial measurements for scalar measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Adjusted Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

> **Note**
>
> If the last contiguous read date/time is not currently set for the measuring component then the initial measurement is assumed to be the first initial measurement received for the measuring component the Last Contiguous Read Date/Time will be set to the initial measurement To Date/Time. Furthermore, even if the Last Contiguous Read Date/Rime does not align to one of the expected measurement date/times, the Last Contiguous Read Date/Time and Adjusted Latest Contiguous Read Date/Time will be updated.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If Auto-Read Scalar Periodic Estimation is being implemented for Auto-Read Scalar Devices that have existing measurement history, then the "No of Hours in Past to Retrieve Last Useable Measurement" option of the "Measurement Data Option" Feature Configuration Type should be configured. Logic will not initialize the latest measurement date/time information for any measuring component that does not have any latest measurement date/time information but does have existing final measurements. In these cases, the setting of the latest measurement date/time is deferred to the Auto-Read Scalar Periodic Estimation process. Please refer to About Periodic Estimation for more information.

If the initial measurement's device configuration is configured for related measuring component consumption synchronization this algorithm can optionally identify whether there are any consumption synchronization eligible measurements on the related measuring component (eligible measurements are defined as those whose condition falls between the bottom and top ranges for consumption synchronization or the additional condition code for consumption synchronization). When the initiating initial measurement is not "back dated" or the synchronization period does not extend past the initial measurement's date range the appropriate interval or scalar consumption synchronization activity will be selected based on the measuring component type of the related measuring component. When the initiating initial measurement is "back dated" and there is a contiguous period of synchronization eligible measurements that extends past the date range of the initial measurement the Gap Consumption Sync Activity will be created.

Lastly this algorithm can be configured to maintain outage related activities for the device. Since the receipt of initial load data for a device indicates it is now powered on and communicating if there are any outstanding outage or estimation suppression activities (initiated via widespread outage) they can be ended as of the from date/time of the initial measurement.

**D1-UPDDTSCMC - Update Latest Date/Time on Scalar Measuring Component**

This algorithm applies to estimation initial measurements for scalar measuring components.

This algorithm maintains important summary level dates on the Measuring Component (MC) as well as initiating the Consumption Synchronization process.

The following dates are maintained:

- Last Contiguous Read Date/Time
- Adjusted Last Contiguous Read Date/Time
- Most Recent Read Date/Time
- Most Recent Non Estimated Read Date/Time

> **Note**
>
> If the last contiguous read date/time is not currently set for the measuring component then the initial measurement is assumed to be the first initial measurement received for the measuring component the Last Contiguous Read Date/Time will be set to the initial measurement's To Date/Time. Furthermore, even if the Last Contiguous Read Date/Time does not align to one of the expected measurement date/times, the Last Contiguous Read Date/Time and adjusted Latest Contiguous Read Date/Time will be updated.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If Auto-Read Scalar Periodic Estimation is being implemented for Auto-Read Scalar Devices that have existing measurement history, then the "No of Hours in Past to Retrieve Last Useable Measurement" option of the "Measurement Data Option" Feature Configuration Type should be configured. Logic will not initialize the latest measurement date/time information for any measuring component that does not have any latest measurement date/time information but does have existing final measurements. In these cases, the setting of the latest measurement date/time is deferred to the Auto-Read Scalar Periodic Estimation process. Please refer to About Periodic Estimation for more information.

As this algorithm is used on estimation initial measurement data it is not common that it will update the Most Recent Non Estimated Read Date/Time. However, there are circumstances where high quality estimations based on a check channel might be given a measurement quality in the regular range. We sometimes refer to these as prorated measurements.

It is also important to note that this algorithm will not initiate consumption synchronization processing.

# Direct Measurement Processing

Direct Measurement Processing is a method of processing incoming initial measurement data designed to reduce the volume of initial measurement records in the database. Rather than create and update initial measurement data records, Direct Measurement Processing processes initial measurement data in memory and creates final measurements in the Measurement (D1_MSRMT) table. With Direct Measurement Processing, records in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table are created only if an error is encountered during processing (including VEE processing). If using Measuring Component Ordering (see below), Direct Measurement Processing also creates temporary staging records (in the IMD Control Staging (D1_IMD_CTRL_STAGE) table).

> **Note**
>
> Direct Measurement Processing is available with Native Implementations only, and only for processing Initial Load measurements. It cannot be used for Manual or Estimation measurements.

> **Note**
>
> The **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts for measurement data processed using Direct Measurement Processing.

### IMD Types

IMD Types are configuration records used to define how initial measurements are processed using Direct Measurement Processing. There are two main categories of IMD types - seeder types (based on the Seeder IMD Type (D1-IMDSeederType) business object) and specific types (based on the Specific IMD Type (D1-IMDType) business object).


Copyright © 2026, Oracle and/or its affiliates.

- **Seeder IMD types** represents an initial measurement data (IMD) seeder, which serves as the universal input into the system for external callers or for initial measurement payload processing. The main purpose of the seeder IMD type is to determine the most appropriate specific IMD type (such as interval or scalar, or a type used with a particular Smart Grid Gateway adapter) to use for further processing and prepare the input data for that IMD type. Additionally, a seeder IMD type is used to perform critical validation (such as an interval count validation), device and channel identification, mapping of external values (such as Unit of Measure and time zone) to internal values, and time zone conversion for date/time fields.

  > **Note**
  >
  > There can only be on seeder IMD type active at any one time.

- **Specific IMD types** define the rules for processing a specific initial measurement data (IMD) record. Different specific IMD types contain rules for processing the different types of initial measurement data that can be received. For example, one IMD type defines the rules for processing a scalar measurement from the Landys+Gyr head end system while another IMD Type instance defines the rules for processing an interval initial measurement from the Sensus head end system. There are four types of algorithms supported for specific IMD types: Pre Processing, Pre-VEE, VEE, and Post-VEE. These algorithms are responsible for validating that the measurement has been correctly mapped to the IMD Type (Pre Processing), calculating consumption (Pre-VEE), performing VEE (VEE), writing to the final measurement table (Post-VEE), among other responsibilities.

IMD Types are defined for specific measuring component types in the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) for each head end system. See Creating Head End Systems in the *Administrative User Guide* for more information about processing methods.

See Understanding IMD Types in the *Administrative User Guide* for more information.

#### VEE Processing and Exceptions with Direct Measurement Processing

When using Direct Measurement Processing, VEE is performed by the Direct Measurement Processing Service based on algorithms defined for the initial measurement's IMD Type.

- If any quantities are changed during VEE processing, original (Pre VEE) quantities are captured along with the final (Post VEE) quantities in the Measurement (D1_MSRMT) table.
- If an initial measurement fails a VEE rule with only exceptions with a Severity of "Information", a header record is created in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table. This header serves as a parent record for the VEE exception(s). In this case, final measurements are created as part of the process.
- If an initial measurement returns an error during Pre-VEE processing or fails a VEE rule with a Severity of "Issues" or "Terminate", a record is created in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table for the initial measurement, and further processing is done using Legacy Measurement Processing. The type of initial measurement data record created is based the configuration of the IMD type.

The table below summarizes how VEE exceptions are created and and whether final measurements are created when using Direct Measurement Processing.


Copyright © 2026, Oracle and/or its affiliates.

| Scenario | Initial Measurement Data (D1_INIT_MSRMT_DATA) Record | VEE Exceptions Created | Final Measurements Created |
|---|---|---|---|
| No exceptions | No | No | Yes |
| Error in Pre VEE algorithms | Yes, based on the IMD Business Object defined on the IMD Type (such as D1-InitialLoadIMDInterval) | No | No |
| "Information" and "Issue" or "Terminate" VEE Exceptions | Yes, based on the IMD Business Object defined on the IMD Type (such as D1-InitialLoadIMDInterval) | Yes | No |
| "Issue" or "Terminate" VEE Exceptions | Yes, based on the IMD Business Object defined on the IMD Type (such as D1-InitialLoadIMDInterval) | Yes | No |
| Only "Information" VEE exceptions | Yes, Informational Exception Header (D1-InfoExceptionHeader) | Yes | Yes |

> **Note**
>
> When using Direct Measurement Processing with Machine Learning Anomaly Scoring, VEE processing is only performed if a measurement is deemed "anomalous" or if a Post-ML VEE Group is defined for the measuring component or measuring component type. Otherwise, VEE processing is skipped. See Machine Learning Anomaly Scoring for more information.

### Measuring Component Filtering

Direct Measurement Processing supports filtering of incoming data for external measuring components. See External Measuring Components for more information about external measuring components.

Measurement data for external measuring components can be filtered (in other words, dropped) or passed through to an external system via file system or object storage.

Filtering is controlled using the **IMD Handling** flag on the external measuring component types. This flag has two values, "Filtered" and "Pass through". As data is processed, the Direct Measurement Processing Service evaluates this flag to determine how to proceed.

- When this flag is set to "Filtered", the Direct Measurement Processing Service drops the measurement records, returns the initial measurement payload to the payload processing service (marked as "Filtered") so it will be captured in payload processing statistics.
- When this flag is set to "Pass through", the Direct Measurement Processing Service returns the initial measurement payload to the payload processing service (marked as "Pass through") so it will be captured in payload processing statistics, and the Payload


Copyright © 2026, Oracle and/or its affiliates.

Processing batch process sends the payload to a file system or object storage (for use in external systems).

In the event of an error when processing initial measurements for "pass through" external measuring components, the measurement can be captured in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table for correction (similar to what happens if an error is encountered during processing of regular measurement data). Errors for "pass through" measurements can include errors related in critical validations during IMD seeder processing, status mapping, or consumptions calculation. Processing of "pass through" measurements in error is handled via "Pass through" IMD types and measurements are captured in "Pass through" initial measurement business objects. When the error is resolved, the measurement enters the "Waiting for Extract" status, and a batch process will retrieve the payload and send it the file system or object storage.

### Measuring Component Ordering

In some implementations, it's important that measurement data is processed in a specific order. For example, when using a scalar channel as a "check channel" to estimate/validate a non-primary interval channel, it's important to process the scalar channel first. Another example are implementations in which measurement data for some (usually daily) scalar measuring components is occasionally sent to the system out of order. For example, data for four days in January might arrive in the following order:

- January 5
- January 2
- January 3
- January 4

By default, Direct Measurement Processing processes initial measurements as they are received into the system, meaning that the data for the January 5 measurement would be updated three times as each of the measurements that came after it is processed.

In cases like these, measurements should be processed based on the initial measurement's Start and Stop dates. In Legacy measurement processing, situations like this are handled by the IMD Monitor batch processes (such as D1-IMD and D1-IMDV2), which processes the data based on Start and Stop dates rather than in the order in which the data was received in the application. For Direct Measurement Processing, this can be supported through Measuring Component Ordering.

Measuring Component Ordering can be used with specific measuring components or measuring component types, and is defied for measuring components or measuring component types using the **Initial Load IMD Ordering** or **Initial Load IMD Ordering (Fallback)** flag (respectively). Initial measurements marked for Measuring Component Ordering via this flag are captured in the IMD Control Staging table with a status of "Ready to Process". Lists of these initial measurements can be viewed in either the **Initial Measurement Data History** zone of the **Measuring Component** portal, or by searching by measuring component or device or IMD identifier using the **Initial Measurement Data Search** zone in the **IMD Query** portal. Individual initial measurements in the "Ready to Process" state can be viewed using the **IMD Control Staging** portal (see Using the IMD Control Staging Portal for more information about using that portal).

"Ready to Process" records can be processed in one of two ways:

- One at a time from the **IMD Control Staging** portal
- Using the Process IMD Control Stage Data (D1-IMDCS) batch process. This batch process uses the following parameters:


Copyright © 2026, Oracle and/or its affiliates.

- **Value to restrict to IMD Control Stage Status**: Designates the status of records to process. When processing Measuring Component Ordering initial measurements, this should be set to "D1RD" ("Ready to Process").
- **Enable sequence handling**: Designates (true or false) if initial measurements should be retrieved and ordered for a measuring component in ascending order by date time and create date time. When processing Measuring Component Ordering initial measurements, this should be set to "true".

**Enabling Direct Measurement Processing**

To enable Direct Measurement Processing, do the following:

1. Enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Configuring Direct Measurement Processing and Creating SGG Payload Processing Extendable Lookup Values in the *Administrative User Guide* for more information.
2. Update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Configuring Direct Measurement Processing and Creating Head End Systems in the *Administrative User Guide* for more information.

**Enabling Direct Measurement Processing with Machine Learning Anomaly Scoring**

To enable Direct Measurement Processing with Machine Learning Anomaly Scoring, do the following:

1. Enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Configuring Direct Measurement Processing and Creating SGG Payload Processing Extendable Lookup Values in the *Administrative User Guide* for more information.
2. Update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Configuring Direct Measurement Processing and Creating Head End Systems in the *Administrative User Guide* for more information.
3. Set the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag to "Yes" on the specific measuring components or measuring component types (respectively) you wish to process using machine learning anomaly scoring.
4. Configure an SGG ML Payload Processing Configuration extendable lookup value, and enable the **Use IMD Type** option. See Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.

- Processing Initial Measurement Data Using Direct Measurement Processing

## Processing Initial Measurement Data Using Direct Measurement Processing

Initial measurement data (IMD) received from external systems is processed through two major stages:

- **Seeder IMD Seeder Type**: This is the common interface point that receives all IMD information and performs critical validations and conversions. During this process the IMD Seeder identifies the appropriate specific IMD business object for further processing. The choice is made based on the category of IMD being processed (initial load) and the type of measuring component the IMD is for. For example, an IMD for an interval measuring component would result in the Initial Load IMD (Interval) business object being selected. If for some reason the IMD Seeder encounters an error an instance of the Legacy IMD


Copyright © 2026, Oracle and/or its affiliates.

Seederbusiness object associated with the IMD Type will be created to manage the resolution of that error.

- **Specific IMD Type**: Each specific IMD type has the appropriate calculation and validation logic for the type of data that has been received. An IMD containing scalar data will be processed differently than interval data. However, all types of data will have a similar set of steps taken:
  - Any external system specific statuses for the measurement data are interpreted into the appropriate internal statuses
  - An initial consumption calculation is performed
  - VEE rules are executed
  - Final measurements are created

> **Note**
>
> When using Direct Measurement Processing with Machine Learning Anomaly Scoring, VEE processing is only performed if a measurement is deemed "anomalous" or if a Post-ML VEE Group is defined for the measuring component or measuring component type. Otherwise, VEE processing is skipped. See Machine Learning Anomaly Scoring for more information.

### Seeder IMD Type Processing

The Seeder IMD type receives IMD data in a common structure, however, the content of the data (identifiers, date/times, and so on) can vary depending on the system and the particular device that generated it. To accurately translate the external content to values understood by the system the seeder leverages the configuration of the service provider (aka head end or external system) and the device.

The service provider specifies the following inputs into seeder type processing:

- **Identifier Types:** Each IMD must identify the device and measuring component the data has been generated from. The system supports a multitude of identifiers for both the device and measuring component. The service provider configuration specifies which identifier types are used for both the device and the measuring component which enables the IMD seeder to search for the correct destination for the measurement data. For example, a service provider might indicate that devices are identified by a serial number and measuring components are identified by a channel identifier.
- **Date/Time Format:** The service provider configuration will identify whether the date/times that were sent included specific time zone designations. This allows the system to accurately handle any time zone conversions that may be necessary.
- **Processing Methods:** A service provider has many processing methods that are used in the evaluation of the IMD.
  - Initial Measurement Creation - Maps a given measuring component type to the initial measurement BO that should be created
  - UOM Translation - Identifies the extendable lookup that should be used to map the service provider specific UOM to the internal UOM
  - TOU Translation - Identifies the extendable lookup that should be used to map the service provider specific TOU to the internal TOU
  - SQI Translation - Identifies the extendable lookup that should be used to map the service provider specific SQI to the internal SQI


Copyright © 2026, Oracle and/or its affiliates.

- Time Zone Translation - Identifies the extendable lookup to be used in mapping the service provider specific time zone to the interval time zone.

The device specifies the following inputs into seeder type processing:

- **Incoming Data Shift:** This value is either specified directly on the device or can be defaulted via fallback on the device type. It identifies whether the incoming date/time information has is "Shifted" meaning it has been adjusted for Daylight Savings Time (DST) when in effect, or it is "Not Shifted" which means it will never be adjusted for DST. This plays an important role in the date/time processing since all incoming IMD date/times will be stored without an adjustment for DST.

With these inputs the seeder type is able to translate the external identifiers into the appropriate identifiers to tie the initial measurement data to a specific measuring component in the system.

Once a measuring component is identified it is then used to select the appropriate specific IMD type using the "Initial Measurement Creation" processing method of the service provider.

#### Seeder IMD Type Critical Validations

For initial measurement data to continue on to specific IMD type processing it must pass several critical validations:

- The measuring component must be identified.
- The initial measurement must contain the appropriate date/times for the type of data being processed: scalar must have an end date/time and interval must have both a start and end date/time.
- The interval list must match the number of intervals expected based on the start and end date/times. Note: if there are too few intervals the missing intervals can be automatically added to the initial measurement without generating an error. How this is done depends on whether there are date/times for each individual interval. If there are date/times for each interval the missing intervals are placed where the gaps exist. Otherwise when there are no date/times for the individual intervals the missing intervals are added to the end of the interval list.

If any of the critical validations have failed the IMD processing will be halted and an instance of the IMD Seeder business object will be created, same as when using legacy processing.

Note: There is special logic that exists for handling the situation where either the start date/time or the IMD end date/time falls on the duplicate hour that occurs when a time zone transitions out of daylight savings time. Typically there are two 1AMs on that day, one is considered to be during DST and the other is not. To ensure that the appropriate hour is selected the seeder type uses the count of intervals and final measurements that exist prior to the initial measurement to identify the appropriate hour.

> **Note**
>
> Additional detail on the logic to derive the service provider, device, and measuring component identification can be found on the D1-DER-SPRMC Algorithm Type..

#### Seeder IMD Type Time Zone Translation and Conversion

To provide the most accurate tracking of measurement data all initial measurement data is stored in a single time zone without an adjustment for Daylight Savings Time (DST). As utilities can have multiple head end systems that each send date/time information in different ways and some utilities operate in multiple time zones the seeder type contains functionality to convert


Copyright © 2026, Oracle and/or its affiliates.

incoming date/times from a source time zone into the time zone of the Oracle Utilities application (Meter Data Management or Customer to Meter), (as indicated by the base time zone)

There are three components to identifying the source time zone of the incoming initial measurement:

- **Service Provider IMD Import Date/Time Format:** When the initial measurement date/times are sent in format that identifies the time zone for each date/time, the date/times will be delivered to the seeder type already converted.
- **Incoming Data Shift:** This indicates whether the date/times will be adjusted for DST while it is in effect. A value of "Always in Local Time" indicates that the date/times are adjusted for DST. A value of "Always in Standard Time" indicates that the date/times are not adjusted for DST
- **Time Zone**: The time zone of the incoming data is identified using the follow order of precedence:
    1. The time zone as indicated by the initial measurement data
    2. The service point time zone
    3. The device configuration time zone
    4. The measuring component time zone
    5. The time zone defined on the **Main** tab on the **Installation Options - Framework** portal

Once the time zone and the data shift of the incoming date/times is identified all date/times will be converted from that time zone and shift into the base time zone in standard time.

Note: Additional detail on the logic for date/time adjustments and validation of undercount and overcount can be found on the D1-DODTTMADJ Algorithm Type.

### Specific IMD Type Processing

When initial measurement data is received from external systems (head end systems) it is processed based on the the specific IMD type defined or the measuring component type on the "Initial Measurement Creation" processing method. IMD types are configured to execute the VEE rules that are associated to the Initial Load VEE role on the measuring component (or measuring component type in the case of a fallback). For adapter specific variations of the IMD type there is logic to map external measurement statuses to measurement conditions.

> **Note**
>
> When using Direct Measurement Processing with Machine Learning Anomaly Scoring, VEE processing is only performed if a measurement is deemed "anomalous" or if a Post-ML VEE Group is defined for the measuring component or measuring component type. Otherwise, VEE processing is skipped. See Machine Learning Anomaly Scoring for more information.

### Scalar Initial Measurement Data Processing

Scalar initial measurement processing works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Scalar Initial Measurement Data Processing for details.


Copyright © 2026, Oracle and/or its affiliates.

### Interval Initial Measurement Data Processing

Interval initial measurement processing works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Interval Initial Measurement Data Processing for details.

### Initial Measurement Data Valid Date Boundaries

Initial measurement data valid date boundaries work in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Initial Measurement Data Valid Date Boundaries for details.

### Deferring Initial Measurement Data Processing

Deferring initial measurement data processing works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Deferring Initial Measurement Data Processing for details.

### Initial Measurement Data Prioritization

Initial measurement data prioritization works in the same way in both Legacy Measurement Processing and Direct Measurement Processing. See Initial Measurement Data Prioritization for details.

### Measuring Component Dates

Measuring component dates are maintained in the way in both Legacy Initial Measurement Data Processing and Direct Measurement Processing. See Maintaining Measuring Component Dates for details.

## Machine Learning Anomaly Scoring

Machine Learning Anomaly Scoring is a method of processing incoming initial measurement data designed to use Artificial Intelligence and Machine Learning neural network models to recognize "normal" versus anomalous measurement values, and only perform Validation, Editing, and Estimation (VEE) processing for measurements deemed to be anomalous. Most measurement data for most customers passes VEE processing without exceptions, but that processing requires considerable computing resources and processing time. Skipping VEE processing for "normal" data greatly reduces the compute resources and processing time needed for measurement processing.

**Notes**:

- Machine Learning Anomaly Scoring is available with Software-as-a-Service (SaaS) cloud Implementations only. It should only be used for processing certain types of Initial Load measurements for Residential customers. It can not be used for Estimation or Manual measurements or for Commercial or Industrial customers.
- Machine Learning Anomaly Scoring is available under Limited and Controlled Availability for select Early Adopters.
- Machine Learning Anomaly Scoring relies on machine learning models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Machine Learning Anomaly Scoring.


Copyright © 2026, Oracle and/or its affiliates.

### The Machine Learning Anomaly Scoring Process

Machine learning Anomaly Scoring can be used with specific measuring components or measuring component types, and is defied for measuring components or measuring component types using the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

Initial measurements for measuring components flagged for Machine Learning (via the **Enable ML Based Validation** flag) are created as usual, but are marked with an Execution Method of "Machine Learning". These measurements are transitioned to the "Waiting for ML" State, and a copy of the measurement is sent to an Object Storage location, where it is picked up by a deep learning/neural network machine learning model. The model processes the data using the incoming measurement data, sample data, and weather data to calculate estimated values for each measurement value in the payload (see **Machine Learning Models**, below for more details), and then returns the updated data to a different Object Storage location. A **Machine Learning** payload processing batch process picks up the payload and updates the initial measurement records with the machine learning measurement values as well as the **Anomaly Score Threshold** specified on either the measuring component, measuring component type, or the MDM Master Configuration. It then calculates the difference between the original Pre-VEE value and the machine learning value for each measurement value in the payload (this is known as an "Anomaly Score") and compares each to the **Anomaly Score Threshold**. If any of the Anomaly Scores for any of the measurement values exceeds the **Anomaly Score Threshold**, the initial measurement is deemed "anomalous", an exception is created, and the measurement proceeds through the normal initial measurement data lifecycle, including VEE processing and creation of final measurements. If none of the Anomaly Scores for any of the measurement values exceeds the Anomaly Score Threshold, the initial measurement moves directly to the creation of final measurements unless Post-ML VEE processing has been configured for the measuring component or measuring component type (see **Post-ML VEE Processing** below).

> **Note**
>
> **Note**: The Anomaly Score Threshold for a measuring component is selected using the following order of precedence: (1) Measuring Component threshold (if present), (2) Measuring Component Type threshold (if present), (3) MDM Master Configuration.

### Anomaly Threshold Considerations

The Anomaly Threshold can be set at three levels:

- Default (via Master Configuration): This is used to set a default threshold for all measuring components that don't have a threshold set for their type or themselves.
- Measuring Component Type: This is used to set thresholds for groups of measuring components that share a type.
- Measuring Component: This is used to set the threshold for each specific measuring component. This is a good approach to set a threshold for any unique measuring component that is idiosyncratic and is either failing with too many false positives or passing with too many false negatives.

There are several consideration to take into account when setting the Anomaly Threshold at each of these levels.

- **Interval vs. Scalar**: Because of the considerable difference between interval values and scalar values, using the same Anomaly Threshold for both interval and scalar measuring


Copyright © 2026, Oracle and/or its affiliates.

components can result in many "false positive" results. Consider a threshold of 10. That would be a reasonable threshold for a single interval but not for a scalar measurement that covers the entire day. To account for this, you might set the default threshold (via the Master Configuration) as the threshold to use for all scalar measuring components, and use measuring component type or measuring component thresholds for interval measurements.

- **Interval Size**: Similar to the difference between interval and scalar values, measurement values for 15-minute intervals are generally smaller than those for 1-hour intervals. As above, using the same threshold for all interval measuring components could result in "false positives". To account for this you might define thresholds based the Interval Size defined for different measuring component types. For example, 15-minute measuring component types might have a threshold of "4", while 1-hour measuring component types might have a threshold in the 13-16 range.
- **Service Type**: Interval and scalar values are likely to differ between commodities and service types. For example, if your system records daily scalar measurements for both gas and electricity, the scalar values for those are likely to be quite different. A household that uses 29 to 30 kilowatt-hours (kWh) of electricity per day might use only 1 to 4 therms of natural gas per day. To account for this, you might use specific measuring component type or measuring component thresholds for gas measuring components and different measuring component type or measuring component thresholds for electric measuring components.

### Machine Learning Models

The neural network machine learning model used with machine learning anomaly scoring is trained on millions of data points to recognize "normal" versus anomalous usage patterns. When the model runs on your customers' data, it processes the incoming measurement data along with weather data to calculate estimated values for each measurement value in the payload.

Machine Learning Anomaly Scoring includes 3 machine learning models used with specific types of data:

- Residential 1 hour interval data
- Residential 15 minute interval data
- Residential daily scalar data

### Post-ML VEE Processing

If needed, you can define VEE Groups that are executed after machine learning processing to compare or adjust the initial measurement based on a related measuring component. These groups are referred to as "Post-ML" VEE groups, and are defined for the **Post-ML** VEE Role on measuring component types and measuring components. Note that Post-ML VEE Groups can only be used when machine learning is enabled for the measuring component or measuring component type via the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

Post-ML VEE Groups can include only the following VEE Rules:

- Sum Check
- Interval Adjustment From Scalar
- Scalar Calculation from Interval
- Final Measurement Replacement


Copyright © 2026, Oracle and/or its affiliates.

See Configure Post-ML VEE Processing in Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.

### Enabling Machine Learning Anomaly Scoring

Enabling Machine Learning Anomaly Scoring involves the following:

1. Set the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag to "Yes" on the specific measuring components or measuring component types (respectively) you wish to process using machine learning anomaly scoring.
2. Set the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** on specific measuring components or measuring component types (respectively) if needed.
3. Configure an SGG ML Payload Processing Configuration extendable lookup value. See Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.

> **Note**
>
> Machine Learning Anomaly Scoring is available under Limited and Controlled Availability for select Early Adopters.
> Machine Learning Anomaly Scoring relies on machine learning models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Machine Learning Anomaly Scoring.

### Enabling Machine Learning Anomaly Scoring with Direct Measurement Processing

To enable Machine Learning Anomaly Scoring with Direct Measurement Processing, do the following:

1. Set the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag to "Yes" on the specific measuring components or measuring component types (respectively) you wish to process using machine learning anomaly scoring.
2. Set the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** on specific measuring components or measuring component types (respectively) if needed.
3. Configure an SGG ML Payload Processing Configuration extendable lookup value, and enable the **Use IMD Type** option. See Configuring Machine Learning Anomaly Scoring in the *Administrative User Guide* for more information.
4. Enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Configuring Direct Measurement Processing and Creating SGG Payload Processing Extendable Lookup Values in the *Administrative User Guide* for more information.
5. Update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Configuring Direct Measurement Processing and Creating Head End Systems in the *Administrative User Guide* for more information.

## CSV Upload

Measurement data for a measuring component can be uploaded via comma separated values (CSV) files. See Uploading Measurement Data for more information.


Copyright © 2026, Oracle and/or its affiliates.

## Importing Data From External Systems

In addition to importing data from head end systems for meters and other devices, the system can also import data from other types of external systems such as weather systems or from market participants such as an independent system operator (ISO) as part of a settlement market. Data received from these types of systems is stored in standalone measuring components (see About Measuring Components for more information).

When importing data for standalone measuring components, initial measurement processing uses the **Standalone Identifier** (or **Standalone ID**) to identify the measuring component for which the measurement data belongs.

Each external system from which data is imported should be defined as an external application. If the data being imported uses units of measure or time of use periods that are specific to the external system, translation methods should be defined to map the specific data to standard codes.

## Using the IMD Control Staging Portal

You use the **IMD Control Staging** portal to view and maintain records in the IMD Control Staging table. You would typically access records using this portal in the following situations:

- With Direct Measurement Processing when a measuring component has been configured for Measuring Component Ordering and an initial measurement is in the "Ready to Process" state (see Direct Measurement Processing and Measuring Component Ordering for more information)

You can access the portal from either the **Initial Measurement Data History** zone of the **Measuring Component** portal, or the **Initial Measurement Data Search** zone in the **IMD Query** portal.

When you select a record in one of those zones, you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **IMD Control Staging Details**: displays basic attributes of the initial measurement.
  - To process an initial measurement from this portal, click **Process** in the **Record Actions** section.
    - Initial measurements in the the "Ready to Process" **Status** are processed using Direct Measurement Processing.
- **IMD Stage Payload**: displays the initial measurement's payload

# Final Measurements

- About Final Measurements

## About Final Measurements


Copyright © 2026, Oracle and/or its affiliates.

Final measurements are measurements that have been validated, and if necessary, edited & estimated, and is ready for use in downstream processing such as bill determinants calculations.

When an initial measurement is considered "final," that is, it has passed all VEE processing and no additional modifications or changes need to be made, it is transformed into a Final Measurement, or simply a Measurement (the terms measurement, final measurement, and final consumption all reference this same "final" measurement data). Only one final measurement can exist for a given date/time for a given measuring component; one final measurement exists per interval, and likewise one final measurement exists for each scalar reading. In both cases, the final measurement value stored represents the amount consumed between its date/time and the prior final measurement's date/time.

When creating final measurements from initial measurement data:

- Final measurements are created using Post VEE quantities
- Each final measurement's condition is copied from the Post VEE condition
- Initial measurements are normalized into final measurements where each final measurement is for a specific date and time.
- Because a single initial measurement may contain many "readings," a separate final measurement is created for each interval in the initial measurement. For example, if an initial measurement contains 24 hours of 15 minute readings, 96 measurements will be created, each with a specific date and time.
- A Combined Multiplier (created from multipliers defined for the measurement’s measuring component type and install event) is stored for each measurement. In addition, other values are stored for measurements of specific types. The Reading Value (which often differs from the measurement value) is stored for scalar and subtractive interval measurements, and the Reading Condition is stored for subtractive interval measurements. These values are used with measurement reprocessing and other recalculation processes in the event that the original initial measurements have been archived or deleted.

The measurement date/time of a measurement represents the end of the period the measurement value covers. How the start of the period is defined varies slightly based on the type of measuring component associated to the measurement:

- **Scalar Measuring Components**: the measurement date/time represents the point at which the reading was taken. The period the measurement covers is defined as the previous measurements date/time through the measurement date/time.
- **Interval Measuring Components**: the measurement date/time represents the end of the interval period. No date/time is stored to represent the beginning of the interval period, rather it is implied by the interval size of the measuring component. For example an interval measuring component with an hourly interval size and a measurement date/time of 01/01/2010 12:00AM represents the measured quantity for the period of 12/31/2009 11:00PM through 01/01/2010 12:00AM.
- For scalar measuring components there exist special circumstances where a final measurement is not eligible for use in processing (e.g. consumption calculation, usage calculation, etc.). To prevent other processes from using these measurements they are marked with a usage flag of "Do Not Use." This is typically the result of the automatic correction of a rollover threshold error when lower quality measurements inaccurate measurements are removed to enable successful calculation of subtractive consumption.


Copyright © 2026, Oracle and/or its affiliates.

### Conditions

Conditions are used to describe the quality of a final measurement and is an important input into many of our processes. During initial measurement data processing these conditions help VEE rules identify the initial measurements that require more advanced validation or estimation. During usage calculations measurement conditions help to inform the billing system whether the billing determinants produced are based on actual data read from a meter or if it has been estimated (or in many cases a combination of the two).

Conditions follow a numeric sequencing that begins at 000000 and goes up to 999999. As the condition code rises numerically so does its perceived quality. Many areas the system take this into account:

- VEE rules use conditions to identify which initial measurement values need to be estimated or adjusted
- Rollover error automatic corrections allow higher quality data to bypass lower quality data in order to achieve a consumption calculation that does not produce an error.
- Usage calculation rules can provide information about the quality of billing determinants based on the underlying measurement data that feeds into those service quantities.

Throughout the system functionality uses condition code ranges to make decisions. In general the ranges are considered to be:

| Condition Range Start | Condition Range End | Description of Range |
|---|---|---|
| 000000 | 199999 | No measurement |
| 200000 | 299999 | Missing Measurement |
| 300000 | 499999 | Estimated measurement |
| 500000 | 999999 | Regular measurement |

The ranges of "No Measurement" and "Missing Measurement" at first glance sound synonymous but there is an important distinction. With conditions that fall within the "No Measurement" range the expectation is that no measurement should exist. Whereas, the "Missing Measurement" range implies that a measurement was expected but never received.

Note: in some circumstances, like rollover threshold errors, the regular range is further divided to include a "Superior" range of 9000000 - 999999 which can be used to select the appropriate reading to be used to resolve the error.

During initial measurement processing the Final Measurement Replacement VEE rule can be used to increase the perceived quality of a condition code. For example, the "No Read - Outage" condition which falls in the "No Measurement" range can be considered higher quality than "Estimated Measurements" for the purposes of measurement replacement.

Refer to About IMD Validations for more information on the Final Measurement Replacement VEE rule.

### Derived Values

Final measurements can record up to 10 derived values in addition to the "as measured" value. The derivation formula for each value on a final measurement is controlled by the measuring component type.

Derived values are not reliant on consumption values, but can also come from factors, historical data, or another source. For example, a set of measurements can be adjusted or converted into other units of measure. In the below example the first derived value is the result


Copyright © 2026, Oracle and/or its affiliates.

of the application of a loss adjustment factor and the second derived value is the result of applying a pre set formula against the loss adjusted value to compute BTU.

| Date/Time | As Measured UOM: CCF SQI: N/A | Loss Adjusted UOM: CCF SQI: N/A | Thermal Units UOM: BTU SQI: N/A |
|---|---|---|---|
| 01/01/2010 3:00PM | 10 | 10.1 | 10.11 |
| 01/01/2010 4:00PM | 15 | 15.15 | 15.165 |
| 01/01/2010 5:00PM | 10 | 10.1 | 10.11 |

Refer to Understanding Measuring Component Types in the Administration Guide for additional information about configuring derivation algorithms.

**Final Measurements and Direct Measurement Processing**

When using Direct Measurement Processing, final measurement may display additional information if any quantities were changed during VEE processing. In this case, original (Pre VEE) quantities and data are captured along with the final (Post VEE) quantities in the following values:

- Pre-VEE Measurement: The Pre-VEE value for the measurement
- Pre-VEE Measurement Condition: The Pre-VEE condition code or the measurement
- Pre-VEE Reading: The Pre-VEE value for reading (used with scalar measurements)
- Pre-VEE Reading Condition: The Pre-VEE condition code for reading (used with scalar measurements)

# Interval Snapshot Mapping


- About Interval Snapshot Mapping
- Maintaining Interval Snapshot Mapping


## About Interval Snapshot Mapping

Interval snapshot mappings are used to map intervals to a particular point in the day, and are used in the snapshot creation process to align a particular interval with the column it belongs to.

The Interval Snapshot Mapping (D1-IntervalSnapMapping) business service is used to create Interval Snapshot Mapping entries based on the installation time zone. This service takes Seconds Per Interval, Start Date/Time, and End Date/Time as inputs.

## Maintaining Interval Snapshot Mapping

You use the **Interval Snapshot Mapping** portal to view and maintain Interval Snapshot Mapping entries.

Refer to About Interval Snapshot Mapping for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Interval Snapshot Mapping**. You are brought to a query portal with options for searching. You can search by:

- Seconds Per Interval



Copyright © 2026, Oracle and/or its affiliates.

- Date/Time From
- Date/Time To

The search results include all of the mapping entries that match the provided Seconds Per Interval (if provided) and that fall within the specified date range (if provided).

### Creating Interval Snapshot Mapping Entries

The Interval Snapshot Mapping (D1-IntervalSnapMapping) business service is used to create Interval Snapshot Mapping entries based on the installation time zone. This service can be invoked from the **Interval Snapshot Mapping** portal.

To create new Interval Snapshot Mapping entries, either select **Main**, then **Settlement**, then **Interval Snapshot Mapping**, and then Add, or click **Add** in the **Interval Snapshot Mapping** portal. The **Create Interval Snapshot Mapping Entries** page opens.

Enter parameters to specify the mapping entries to create, including:

- Seconds Per Interval
- Date/Time From
- Date/Time To

Click **OK**.

Enter appropriate parameters in the search portal to view the new entries.

# Reader Remarks

Initial measurements for manually-read scalar meters can create reader remarks. Reader remarks are a specific type of device event used to capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters.

Reader remark types are submitted with scalar initial measurements when received from a head end system or meter read collection system. Reader remarks are NOT uploaded along with other device events. Reader remarks are always associated with a scalar initial measurement.

Reader remarks can be passed onto to other subscribing systems such as a customer information system (such as Oracle Utilities Customer Care and Billing), an outage system (such as Oracle Utilities Network Management System), or some other application.

Like other device events, reader remarks can create service issue monitors.

> **Note**
>
> Refer to About Service Issue Monitors for more information on Service Issue Monitors.

### Creating Reader Remarks Manually

Use this procedure to create a new reader remark for an existing initial measurement.

Reader remarks must always be associated with an initial measurement.

Reader remarks are most often imported into the application with an initial measurement. However, users can also create reader remarks using the following procedure.


Copyright © 2026, Oracle and/or its affiliates.

1. Search for and select the initial measurement that the new reader remark is to be associated with.
2. Click the **Add** link on the **Reader Remarks of Initial Measurement** zone on the **Initial Measurement** portal.
3. Select the reader remark type for the reader remark from the **Reader Remark Type** drop-down list.
4. *Optional*. If you need to create more than one reader remark for the initial measurement, click the plus sign, and select the reader remark type for the reader remark from the Reader Remark Type drop-down list. Repeat as needed.

   Note that only a single reader remark of any given type can be associated with a single initial measurement.
5. Click **OK**. The new reader remark is displayed in the **Reader Remarks of Initial Measurement** zone on the **Initial Measurement** portal.

# Periodic Estimation


- About Periodic Estimation


## About Periodic Estimation

Periodic estimation is a process that scans eligible devices and their measuring components for missing measurements that have not been received within a required amount of time. Once missing measurements are identified it then generates estimation initial measurements to fill in those missing measurements. It is important to note that the periodic estimation itself does not actually perform any estimations. It relies on the estimation initial measurement to generate the estimate through the estimation VEE rules that are executed.

The periodic estimation process is as follows:

1. Determine if measuring component is eligible for estimation.
2. Scan the eligible period for missing measurements.
3. Confirm missing measurements do not fall within a period where the device was not installed or was disconnected.
   - For scalar measuring components, initial measurements are NOT created for disconnected periods.
   - For interval measuring components, initial measurements will be created for disconnected periods, but the intervals will be marked as 'No Read - Disconnected' (Measurement Condition code 102000) and will be ignored for estimation by VEE processing.
4. Confirm there are no in progress initial measurements that cover the period of the missing measurements.
5. Either generate an initial measurement or a To Do for each measurement gap that is found.

This process relies on an important date that is maintained on each measuring component:

- **Latest Contiguous Measurement Date/Time**: this represents the last time the measuring component had a complete history of measurements without any gaps. By nature it implies that the is a gap immediately after it.



Copyright © 2026, Oracle and/or its affiliates.

This date allows the periodic estimation process to efficiently identify those measuring components that are missing measurements and need to be estimated.

> **Note**
>
> It is possible to disable periodic estimation for a given device by updating the periodic estimation eligibility on the service point it is associated to. This is useful if there are one-off service points that should either never be eligible for estimation or should temporarily have estimation turned off.

#### Periodic Estimation Feature Configuration

The estimation begin date for interval and scalar periodic estimation can be influenced by a feature configuration option under the feature type Measurement Data Options (D1MD).

When a **No of Hours in Past to Retrieve Last Useable Measurement** option is defined it will have two impacts on periodic estimation, both of which are only applicable when the measuring component's type does not have a value configured for the *Maximum Days to Estimate*.

1. When periodic estimation is run for a measuring component that does not yet have a **Last Contiguous Measurement Date Time** then this option will be used to compute the beginning of the search range to find a useable measurement. That date is then used to begin estimation from.
2. When running for a measuring component where the **Last Contiguous Measurement Date Time** is already set then this option will be used to compute the oldest date that estimation can begin from.

For both scenarios the begin date time calculation will be performed by subtracting the value configured from the current Process Date Time. For example, if this option were set to 744 hours (31 days) and the process date time were 02/01 12:00:00AM then the begin date time would be calculated as 01/01 12:00:00AM.

When implemented this option will also instruct the Update Latest Date/Time on Scalar Measuring Component algorithm type (D1-UPDDTSCMC), which is responsible for maintaining the latest measurement date time information for scalar measuring components, to skip initializing the latest measurement date/time information when an initial measurement is received for a measuring component where there is existing measurement data history and this feature configuration option is configured. For those scenarios the initialization of the latest measurement date/time information should be deferred to the Auto-Read Scalar Period Estimation process. Note: this only applies for situations when the latest measurement date/time information has not been set previously and the initial measurement is not the first measurement received for the measuring component.

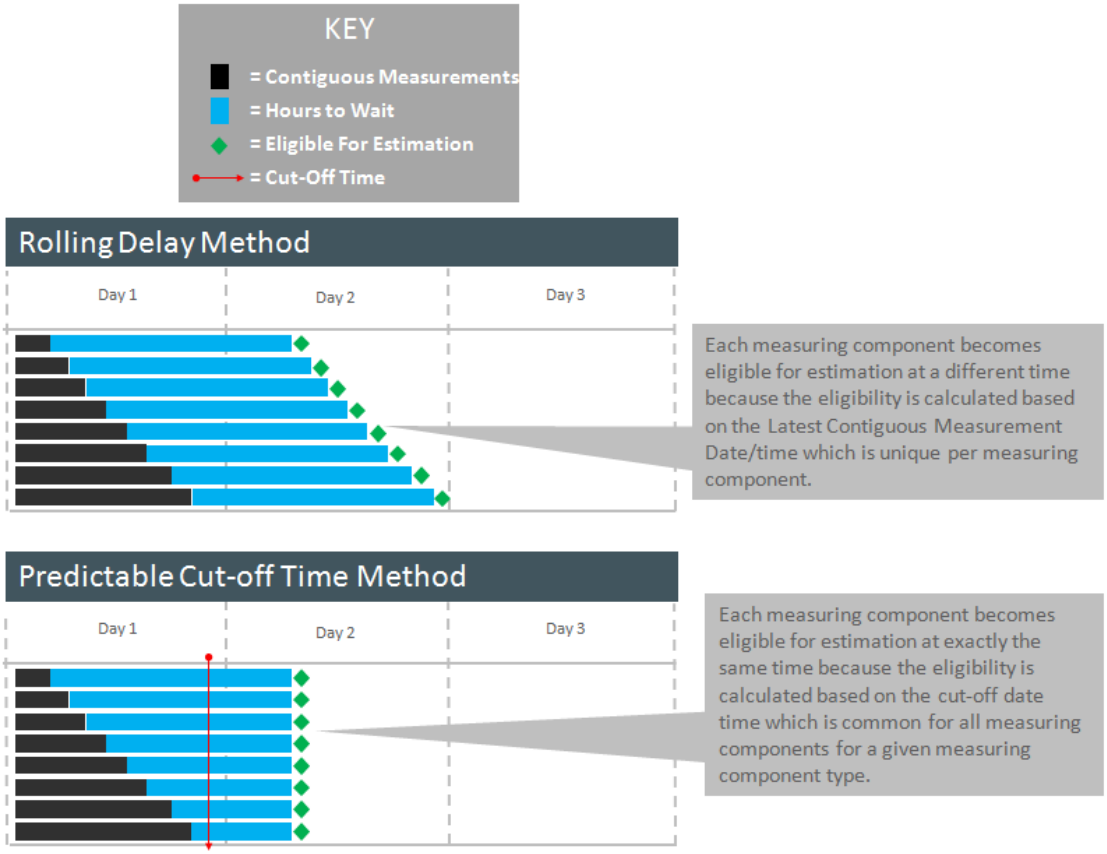
> **Note**
>
> This does not apply to the analogous logic for an interval initial measurement.

#### Interval Periodic Estimation

When missing measurements have been identified by the interval periodic estimation functionality an estimation initial measurement will be created for each contiguous set of missing measurements in the estimation period.

There are two methodologies available for estimating interval measuring components:


Copyright © 2026, Oracle and/or its affiliates.




- **Rolling Delay**: In this method the execution time of the periodic estimation process is an important factor in which measuring components will estimate how far each measuring component estimate. This method supports a minimum number of hours that will be estimated (as defined by the "Hours to Estimate" on the measuring component type).
- **Predictable Cut-off Time:** In this method all measuring components for a measuring component type will estimate until a specified point in time (as defined by the "Cut-off Time" on the measuring component type). With this method the cut-off time determines how far estimations are made rather than the execution time of the periodic estimation process. There is no minimum estimation size for this method it will always estimate up until the cut-off date/time even if that is only a single interval.

The cut-off time takes into consideration daylight savings time (DST) when the device is configured as sending data in legal time (meaning the Incoming Data Shift has been set to Always in Local). In these circumstances the cut-off time will be considered to be adjusted for DST. This is important because measurement data is stored in standard. For example:

| Cut-off Time: 12:00AM | Measurement Time in Standard |
|---|---|
| 01/15/2010 12:00AM (DST not in effect) | 01/15/2010 12:00AM |
| 04/15/2010 12:00AM (DST in effect) | 04/14/2010 11:00PM |

Furthermore, the cut-off time defined on the measuring component type does not have a time zone associated to it and as such it is considered to be defined in the time zone of the particular measuring component being processed. For example, a cut-off time of 12:00AM would be considered to be 12:00AM UTC -08:00 and 12:00AM UTC -05:00 based on the time zones of the respective measuring components. This allows a single measuring component to apply to many time zones and behave uniformly.


Copyright © 2026, Oracle and/or its affiliates.

**Algorithm Type – Create Interval IMD and To Do Based Upon Install History (D1-CIITBIH)**

Algorithms based on this algorithm type determine if a measuring component is missing final measurements during the estimation horizon for any period where the associated device configuration was installed. If final measurements are missing it will either create an estimate initial measurement, create a To Do Entry, or both.

There are a few scenarios that will result in this algorithm being used:

1. If the measuring component's service point has an **Estimation Eligibility** of Not Eligible
2. If the measuring component's service point has open Estimation Suppression activities that overlap the System Date/Time and the MDM Master Configuration field **Fill Missing Data with Zero** is not set to "Yes"
3. If the measuring component's **Latest Contiguous Measurement Date Time** is greater than the process date time less the measuring component type's **Number of Hours to Wait**

The estimation horizon begin date is determined by first determining the initial horizon start date time:

- If the Measuring Component Type's **Maximum Days to Estimate** is configured it will be the more recent of the following two dates:
  - The process date time less the **Maximum Days to Estimate**
  - The Measuring Component's **Latest Contiguous Measurement Date Time**
- Otherwise, if the Measurement Data Option feature configuration type is configured with a value set for the **No of Hours in Past to Retrieve Last Useable Measurement** option then the start will be calculated as follows:
  - If the **Latest Contiguous Measurement Date Time** is populated then take the more recent of:
    - The process date time less the **No of Hours in Past to Retrieve Last Useable Measurement**
    - The Measuring Component's **Latest Contiguous Measurement Date Time**
  - Otherwise, if the **Latest Contiguous Measurement Date Time** is not populated then:
    - Search for a measurement between the following two dates:
      - Start: the process date time less the **No of Hours in Past to Retrieve Last Useable Measurement**
      - End: the process date time
    - If a measurement is found that date is used. Otherwise the process date time less the **No of Hours in Past to Retrieve Last useable Measurement** is used.
- Otherwise, the Measuring Component's **Latest Contiguous Measurement Date Time** is used.
- Otherwise, if the device is installed it will be set to the **Installation Date Time** of the Installation Event (IE).

The estimation horizon end date will be calculated based on the measuring component type's **Periodic Estimation Execution Method**.

**Rolling Delay**:

The more recent of the following dates will be used:


Copyright © 2026, Oracle and/or its affiliates.

- The **Latest Measurement Date Time** plus the Measuring Component Type (MC Type) - **Number of Hours to Estimate** defined.
- The **Process Date Time** less the MC Type - **Number of Hours to Wait**
- Estimations are limited by either a device configuration change or removal. Therefore, the date determined above will be limited to the earlier of the device configuration **Effective End Date Time** or the **Removal Date Time** on Install Event.

**Predictable Cut-off**:

In predictable cut-off the intent is to only estimate up to the last cut-off time that has passed based on the **Number of Hours to Wait** from the measuring component type and no further.

First the process date time will be adjusted based on the **Number of Hours to Wait** configured on the measuring component type. For example, if the process date/time is 01/15/2010 06:00PM and the **Number of Hours to Wait** is 48 then the adjusted process date/time is: 01/13/2010 06:00PM.

Next, the last cut-off time to have passed will be identified and that will become the estimation end date time. In our above example this would be 01/13/2010 12:00AM.

Once the estimation horizon begin and end date times have been identified they may be further split into smaller periods based upon the Measuring Component's device configuration installation history.

For Example:

- Estimation horizon is April 10th at 12AM through April 20th at 12AM
- The Measuring Component's device configuration was installed at one service point on April 1st at 12AM and removed on April 15th at 12AM
- The Measuring Component's device configuration was then installed at a new service point on April 18th at 12AM
- Given the above information the estimation horizon would be broken up into the following two periods:
  - April 10th 12AM through April 15th 12AM (this is for the first installation)
  - April 18th 12AM through April 20th 12AM (this is for the second installation)

Each of the estimation horizon periods determined based upon the installation history will be further examined to determine contiguous periods where final measurements are missing.

For Example:

- Processing April 10th through April 15th
- One final measurement exists for April 12th for 8AM through 9AM
- Yields the following sub-periods:
  - April 10th 12AM through April 12th 8AM
  - April 12th 9AM through April 15th 12AM
- -For every missing period found,
  - Find Initial Measurements that are in a non Final state and the initial measurement's period has an overlap with the missing period.
    - For each initial measurement found,


Copyright © 2026, Oracle and/or its affiliates.

* If the initial measurement's period falls exactly within the boundaries of the missing period, skip the creation of an estimate initial measurement for the missing period.
* If the initial measurement's end date/time is before the start of the missing period, the initial measurement's end date/time is now the start of the missing period. Proceed to create an estimate initial measurement for the adjusted missing period.
* If the initial measurement's start date/time is after the start of the missing period, the initial measurement's start date/time is now the end of the missing period. Proceed to create an estimate initial measurement for the adjusted missing period.

Each time Periodic Estimation is executed the **Latest Contiguous Measurement** will be validated and potentially updated. For example, if it is before the install event's **Installation Date Time**, it will be updated to be the install event's **Installation Date Time**.

> **Note**
>
> If the **Latest Contiguous Measurement Date Time** falls within the installation window of a prior installation it will not be adjusted to the current install event's **Installation Date Time**.

### Scalar Periodic Estimation

Scalar periodic estimation is only available for automatically read scalar measuring components since it requires that scalar measuring components be read on at least a daily basis to be eligible.

The scalar estimation process relies heaving on details from the measuring component type to identify when a measurement is missing. This is because scalar measurements are not required to fall on specific boundaries like interval data. For example, a scalar measuring component read on a daily basis around midnight could have measurements at 12:00AM, 11:55PM, 12:01PM or even at various other points during the day if a field reading or on demand read is performed.

Auto-read scalar measuring component types are configured to provide information that will:

- Identify at what point or points during the day the measuring component will be read. This is a combination of the Hours Between Measurements and the First Daily Measurement Time on the measuring component type.
- Identify the allowable tolerance for measurements that do not fall on the specific expected measurement date/times (e.g. 11:55PM and 12:05AM are OK but 9AM is not). This is defined by the Early and Late Measurement Threshold on the measuring component type.

This allows the scalar periodic estimation process to identify the points during the day that measurements are expected and if no measurement are found it will generate an estimation initial measurement.

Unlike interval periodic estimation each missing measurement will be created as a separate estimation initial measurement.

The Early Measurement Threshold and Late Measurement Threshold fields support situations where a head end system is not be able to report their readings at a consistent time each day. For example a head end system might send reads as follows:

- 3/1 11:55PM


Copyright © 2026, Oracle and/or its affiliates.

- 3/3 12:01AM
- 3/4 12:00AM
- 3/5 12:02AM
- 3/6 11:59 PM

These fields create a window prior to and after an expected reading where anything received within that window would satisfy the expected reading. For example:

- For a measuring component with an Early Measurement Threshold of 00:10:00 (10 minutes), a reading for 3/1 11:55PM would satisfy the early threshold for the 3/2 12:00AM expected reading.
- For a measuring component with an Late Measurement Threshold of 00:10:00 (10 minutes), a reading for 3/3 12:01AM would satisfy the late threshold for the 3/3 12:00AM expected reading.

To determine if a measuring component is eligible for estimation the process will evaluate if the Last Contiguous Measurement Date/Time far enough in the past. Specifically the number of hours elapsed since the Last Contiguous Measurement Date/Time must be greater than or equal to the Hours Before Estimation from the measuring component type.

> **Note**
>
> The first daily measurement time takes into consideration daylight savings time (DST) when the device is configured as sending data in legal time (meaning the Incoming Data Shift has been set to Always in Local). In these circumstances the first daily measurement time will be considered to be adjusted for DST. This is important because measurement data is stored in standard. For example:

| First Daily Measurement Time: 12:00AM | Measurement Time in Standard |
|---|---|
| 01/15/2010 12:00AM (DST not in effect) | 01/15/2010 12:00AM |
| 04/15/2010 12:00AM (DST in effect) | 04/14/2010 11:00PM |

Furthermore, the first daily measurement time defined on the measuring component type does not have a time zone associated to it and as such it is considered to be defined in the time zone of the particular measuring component being processed. For example, a first daily measurement time of 12:00AM would be considered to be 12:00AM UTC -08:00 and 12:00AM UTC -05:00 based on the time zones of the respective measuring components. This allows a single measuring component to apply to many time zones and behave uniformly.

**Algorithm Type – Auto-Read Scalar Periodic Estimation (D1-ARSPE)**

Algorithm based on this algorithm type determine if a measuring component is missing final measurements during the estimation horizon for any period where the associated device configuration was installed. If final measurements are missing it will either create an estimate initial measurement, create a To Do Entry, or both.

There are a few scenarios that will result in this algorithm being used:

1. If the measuring component's service point has an **Estimation Eligibility of Not Eligible**
2. If the measuring component's service point has open Estimation Suppression activities that overlap the System Date/Time and the MDM Master Configuration field **Fill Missing Data with Zero** is not set to "Yes"


Copyright © 2026, Oracle and/or its affiliates.

3. If the measuring component's **Latest Contiguous Measurement Date Time** is greater than the process date time less the measuring component type's **Number of Hours to Wait**

The estimation horizon begin date is determined by first determining the initial horizon start date time:

- If the Measuring Component Type's **Maximum Days to Estimate** is configured it will be the more recent of the following two dates:
  - The process date time less the **Maximum Days to Estimate**
  - The Measuring Component's **Latest Contiguous Measurement Date Time**
- Otherwise, if the Measurement Data Option feature configuration type is configured with a value set for the **No of Hours in Past to Retrieve Last Useable Measurement** option then the start will be calculated as follows:
  - If the **Latest Contiguous Measurement Date Time** is populated then take the more recent of:
    - The process date time less the **No of Hours in Past to Retrieve Last Useable Measurement**
    - The Measuring Component's **Latest Contiguous Measurement Date Time**
  - Otherwise, if the **Latest Contiguous Measurement Date Time** is not populated then:
    - Search for a measurement between the following two dates:
      - Start: the process date time less the **No of Hours in Past to Retrieve Last useable Measurement**
      - End: the process date time
    - If a measurement is found that date is used. Otherwise the process date time less the **No of Hours in Past to Retrieve Last Useable Measurement** is used.
- Otherwise, the Measuring Component's **Latest Contiguous Measurement Date Time** is used.
- Otherwise, if the device is installed it will be set to the **Installation Date Time** of the Installation Event (IE).

The estimation horizon end date will be whichever of the following dates that is furthest in the past:

- Process Date/Time - **Hours Before Estimation** + **Number of Hours to Estimate**
- The device configuration's effective end date/time
- The Off date/time of the device configuration's installation event.

Second, the algorithm identifies any expected measurement date/time without a final measurement or an in progress initial measurement. For each missing measurement an Estimate initial measurement will be created. Estimations will only be made for periods where the device is installed at a service point. If the **Evaluate Installation Event Off Period** parameter has been set to 'N' or left blank then the device must also be turned on.

To determine if there are any missing final measurements during the estimation horizon the algorithm uses the concept of expected measurement date/times as explained above. If there are any expected measurements not covered by a final measurements as dictated by the early and late thresholds then an initial measurement will be generated.


Copyright © 2026, Oracle and/or its affiliates.

# Measurement Reprocessing


- About Measurement Reprocessing
- Reviewing Measurements for Reprocessing


## About Measurement Reprocessing

One major problem utilities face today is when devices are installed with an incorrect consumption multiplier. When this occurs, the billing consumption and the bills are incorrect retroactively.

The Measurement Reprocessing module automatically corrects historical consumption when a change is detected for either a Measuring Component multiplier (the **Register Multiplier** for register measuring components or **Channel Multiplier** for interval and auto-read register measuring components) or an Install Event **Installation Constant** or **Installation Date/Time**. The system monitors for these changes and automatically reprocesses historical measurements if one of the values changes. When a Measuring Component has been triggered for Measurement Reprocessing, a new message will be recorded on the **Log** tab for the Measuring Component. The message will look similar to this example: "Measuring Component Multiplier updated from 1.000000 to 2.000000. Measurements will be examined for reprocessing". Also, a link to the activity created to perform the reprocessing will be shown in the Related Object column.

Two different Activities are leveraged to perform this reprocessing:

- **Measurement Reprocess Activity - Interval**: this activity reprocesses interval data by dividing existing measurements by the Combined Multiplier stored on each measurement record and creates new System IMDs.
- **Measurement Reprocess Activity - Scalar**: this activity reprocesses scalar data by creating new System IMDs from the Reading value stored on each measurement.

The new System IMDs that are generated will skip both VEE and Consumption Sync but will apply the new multiplier and installation constant. Once the activity is finished, a list of the spawned IMDs will be displayed on the Activity screen.

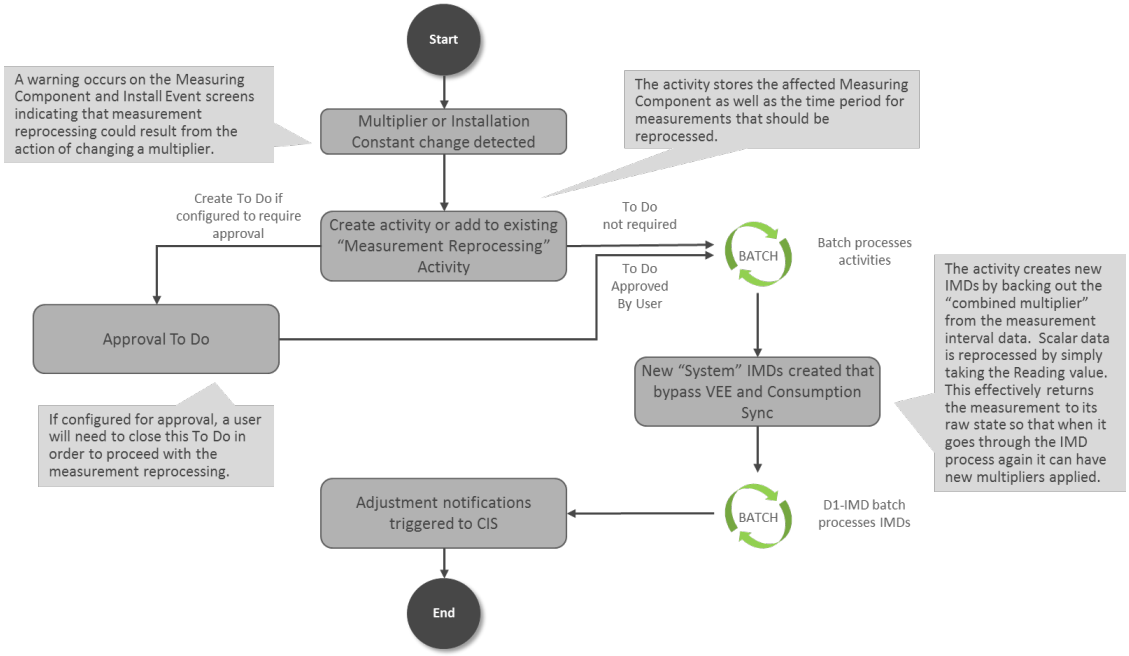
This process works in tandem with the Adjustment Notification process to trigger billing adjustment notifications to the CIS. If usage transactions have already been fulfilled once measurements are reprocessed, then each affected period will be sent to CIS to indicate that an adjustment is required.

If desired, an approval step can be set as required on the related Activity Type. If set, a To Do will be generated for review. The review steps for this To Do are covered in the Reviewing Measurements for Reprocessing section.

The diagram below helps to provide an overview of Measurement Reprocessing:



Copyright © 2026, Oracle and/or its affiliates.




For information on configuration, reference the related section for Configuring Measurement Reprocessing in the *Administrative User Guide*.

## Reviewing Measurements for Reprocessing

When the Measurement Reprocessing Activity Types are configured as "Yes" in the "Approval Required?" field, then a To Do will be generated for user review. The To Do Type for this is set on the Activity Type as well (the configuration for this is covered in detail in the section for Configuring Measurement Reprocessing in the *Administrative User Guide*).

This To Do provides an approval step for a user to examine the Measurement Reprocessing Activity and perform the following:

1. Determine if the multiplier or date change appears legitimate and accurate. Information on the type of change that occurred can be found on the **Log** tab for the Activity. If the change should be rejected, this can be performed using the **Discard** button.
2. Ensure that the period being examined for reprocessing (shown as the Activity Start Date and End Date) covers the period expected. If not accurate, the dates can be changed using the **Edit** button.

If all information on the Activity is correct, then press the **Approve** button to allow reprocessing to continue. Once the activity is finished, a list of the spawned IMDs will be displayed on the Activity screen.

Refer to About Measurement Reprocessing for additional functional information.

# Consumption Sync

- Introduction to Consumption Sync


Copyright © 2026, Oracle and/or its affiliates.

# Introduction to Consumption Sync

Traditional interval and smart meters are often configured with two channels of consumption data, one interval and the other scalar. Utilities configure their meters this way for several reasons, including:

- Two channels provide for a redundant reading in case of a failure.
- Two channels provide for an independent check reading to ensure the meter is working properly.
- Customers often need both types of data for their business processes.

When capturing and storing usage for smart meters with related measuring components, one potential challenge is keeping measurements from related scalar and interval channels synchronized. Measurements for interval and scalar channels for the same device may fall out of alignment for many different reasons, such as:

1. Some meters use different technologies for recording interval and register readings, so the physical readings can be different
2. Readings can vary because of different measurement times
3. Estimation processes can vary by channel and produce different results
4. There can be a mix of actual and estimated measurements

Having two sources of consumption data can cause many problems for utilities. Unfortunately, utilities often need both the scalar and interval data to satisfy their business requirements. For example, scalar register readings may be used for billing, while the interval data is used on a customer self-service portal.

Keeping measurement data aligned between two related measuring components is achieved in several ways:

- During the initial load of initial measurement data VEE rules estimate for missing measurements using measurement data from related measuring components when available. Please refer to About IMD Estimations for more information.
- When measurement data is missing for one or both of the related measuring components the periodic estimation process will generate estimation initial measurement data that will estimate one measuring component then estimate the second measuring component such that the second measuring component's data can be synchronized with the first. Please refer to About Periodic Estimation for more information.
- When new, more accurate, measurement data is received for only one of the measuring components the consumption synchronization process will be initiated to align the previously estimated data with the newly received data. Note: this process is what this section will be concentrating on.

Consumption synchronization is a process by which measurement data for related measuring component is kept synchronized by re-estimating existing measurements when measurement data for related measuring components is received.

For example, consider the following scenario:

- A meter has both scalar and interval channels, and the scalar register channel is considered the "primary" or more reliable channel. The total consumption for each channel should match the consumption for the other channel.


Copyright © 2026, Oracle and/or its affiliates.

- For a particular day, the daily reading for the "secondary" (interval) channel is received before reads from the scalar register have been received. This reading contains two missing intervals, which are estimated during VEE.
- After the interval reading has been finalized, a scalar register reading for the same day is received. The total of this reading is slightly different than the total of the interval reading due to the estimation of the missing intervals.
- The consumption synchronization process can trigger a re-estimation of the two estimated intervals to adjust the total of the interval reading to match that of the scalar register reading.

Other examples of situations that can be resolved using the consumption synchronization process include the following:

- Register Readings whose start/stop times do not align with interval boundaries
- A broken meter for which estimated readings are created for several days
- Requirements that readings from both channels match exactly even if neither contains estimated measurements

### Primary Versus Secondary Measuring Components

When measuring components are related, one is considered the "primary" measuring component, and the other is considered the "secondary" measuring component. How the consumption synchronization process works for a given meter is based largely on which measuring component is considered primary and which is considered secondary.

The following are important principles regarding the primary/secondary measuring components:

1. The primary measuring component has the more reliable data.
2. The secondary measuring component should validate that its consumption aligns with the primary measuring component.
3. If the secondary measuring component needs to be estimated, it should estimate using the available data from the primary measuring component.
4. When possible, initial measurements for the primary measuring component should be processed prior to those for the secondary measuring component (Note: the D1-IMD batch process will process initial measurements in this order). In some scenarios when the secondary measuring component is read more frequently than the primary measuring component, it can be advantageous to violate this principle in favor of being able to execute smaller more frequent loads of measurement data.
5. Periodic estimation should be performed on the primary measuring component first, and then the secondary measuring component.

### Understanding Consumption Synchronization Processing

This section provides an overview of the process that supports keeping consumption for related measuring components synchronized.

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

1. An initial measurement is received (via initial load or manual) for a measuring component whose device configuration is consumption synchronization enabled.


Copyright © 2026, Oracle and/or its affiliates.

2. The consumption synchronization process evaluates the related "consumption reference measuring component" to identify any measurements for the time period of the initial measurement are consumption sync eligible (typically estimated).

   If eligible measurements are found a consumption synchronization activity is created. Note: the activity will not immediately process it will remain in an initial status until a background process is executed.

3. When the consumption sync background process is executed the activity initiates the re-estimation process by creating estimation initial measurements for each measurement that must be synchronized:

   - For scalar measuring components, this initial measurement contains no measurement value.
   - For interval measuring components, this initial measurement can contain a combination of "Regular" measurements (based on existing final measurements) and measurements that require estimation (based on and corresponding to the measurements with condition codes that fell between the range defined for "consumption synchronization" in step 2).

4. Each estimation initial measurement that was created will execute the estimation VEE group rules to recalculate the synchronization eligible measurements.

# About Interval Prices and Price Sets

Interval price sets represent set of interval pricing values used in various types of calculations, including event settlement transactions used with Digital Asset Management, settlement calculations used with Market Settlements Management, or pricing calculations used in billing or rating. Interval price values can include market prices or real-time prices. When using multiple sources of interval prices, each source might be represented by a different interval price set.

Interval prices are individual prices based on a defined interval size (for example, hourly prices).

Interval price are defined by the following data:

- **Interval Price Set Type**: The interval price set type on which the interval price set is based. The interval price set type defines a number of attributes of the price set, including the interval size for the price intervals (such as 1 hour, 15 minutes, and so on), and the unit of measure, time of use period, and service quantity identifier used to define the price values.
- **External Identifier**: An identifier for the interval price set used in external systems
- **Time Zone**: The time zone associated with the interval price set
- **Interval Prices**: A set of interval price values belonging to the interval price set, including the following:
  - **Price Date/Time**: The date and time of the interval price
  - **Price**: The price value (based on the unit of measure defined for the interval price set's type)

**System Data**

Interval Price Sets are based on the following system data:


Copyright © 2026, Oracle and/or its affiliates.

| Data Type | Description |
|---|---|
| Maintenance Object | Measuring Component (D1-MEASRCOMP) |
| Business Object(s) | Interval Price Set (D1-IntervalPriceSet) |

Interval Prices are based on the following system data:

| Data Type | Description |
|---|---|
| Maintenance Object | Measurement (D1-MSRMT) |
| Business Object(s) | Interval Prices (D1-IntervalPrices) |

# Maintaining Interval Price Sets

You use the **Interval Price Set** portal to display and maintain interval price sets and interval prices

Refer to About Interval Price Sets for more information.

You can access the portal from **Menu**, select **Interval Price**, then select **Interval Price Set**. You are brought to a query portal with options for searching for a specific record. Once an interval price set has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Interval Price Set**: This zone displays the selected interval price set.
- **Interval Prices**: This zone displays the individual interval prices for the interval price set.

### Maintaining Interval Prices

You use the **Interval Prices** zone to maintain interval prices for an interval price set. You can either manually add or upload prices.

To manually add prices, click **Add** in the upper right corner of the zone. The **Interval Prices Add** dialog opens. The **Interval Price Set ID** is prepopulated based on the interval price set.

- Enter the **Price Date/Time** and **Price** values for each interval price in the **Interval Prices** section.
- Click **Save** to save the interval prices.

To upload prices, click **Upload** in the upper right corner of the zone. The **Interval Prices Upload** dialog opens. The **Interval Price Set ID** and **Target Time Zone** are prepopulated based on the interval price set.

- Select the **Source Time Format** (Local or Standard) for the price data.
  - **Local** : Date/time values adjusted to "Standard" time (reflecting DST if applicable) and both are captured on the Measurement table
  - **Standard**: Date/time values are captured in both Standard and Local time on the Measurement table
- Specify the **Source Time Zone** for the price data. This is the time zone from which the price data originated. This allows for conversion to the time zone defined for the interval price set.


Copyright © 2026, Oracle and/or its affiliates.

- To upload prices, click **Choose File** to search for and select a comma separated values (CSV) file containing price data and click **Submit**. The **Interval Prices Upload** dialog will display the price data contained in the file. Note that you can edit the price data before saving it.
- Click **Save**to save the interval prices.

> **Note**
>
> You can add prices using both manual entry and upload, but uploading data will overwrite existing prices for the same date/time.

To update one or more prices, select the price(s) to update and click **Edit**. The **Price** dialog opens displaying the selected prices. Edit the prices as needed and click **OK**. Click **Cancel** to close the dialog without making any changes.

To delete one or more prices, select the price(s) to delete and click **Delete**. The **Price** dialog opens asking you to confirm the deletion. Click **OK** to delete the selected prices. Click **Cancel** to close the dialog without deleting the prices.

**Interval Prices File Format**

The file format used to upload interval prices are comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column must included in the first row of the input file.

| Field | Data Type | Sample Content | Description |
|---|---|---|---|
| dt | DATE | 2023-02-11-12.00.00 | Date and time for the interval price |
| value | NUMBER | 0.35 | Interval price amount |

Sample File

The following is a sample file for interval prices spanning a six hour event.

```
dt,value 2023-02-11-12.00.00,0.35 2023-02-11-13.00.00,0.35
2023-02-11-14.00.00,0.35 2023-02-11-15.00.00,0.35 2023-02-11-16.00.00,0.35
2023-02-11-17.00.00,0.35
```

# VEE


- Introduction to VEE
- Validations
- Estimations
- Editing


## Introduction to VEE



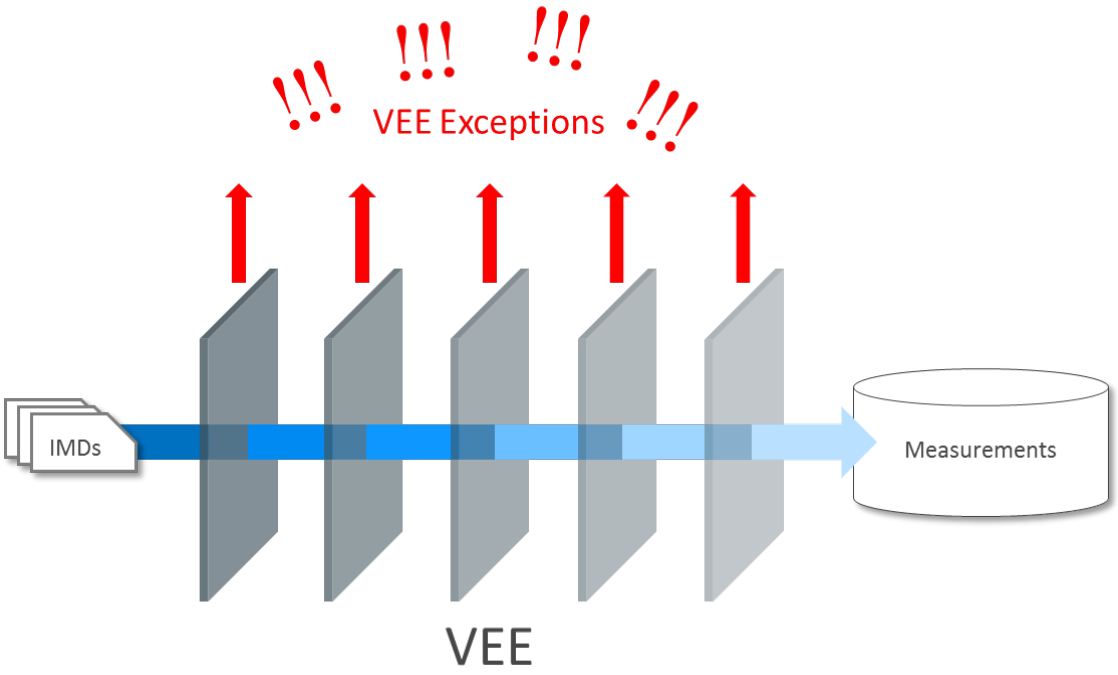
Copyright © 2026, Oracle and/or its affiliates.

- About VEE
- Decision-Making VEE Rules


# About VEE

As measurements are received into Oracle Utilities Meter Data Management there is a need to ensure the data is accurate, fill in any missing data with estimations, and flag any failures for review by users. The VEE (Validation, Estimation, and Editing) process is the primary method for ensuring that these objectives are met.

VEE acts as a multi-layered filtration system for IMDs. Raw data is received from the Head End as Initial Measurement Data and only clean, validated data is received on the other end in the Measurement table. The IMD has both a Pre-VEE and Post-VEE section that details any data changes made as part of VEE.




Initial measurement data can come from different sources, such as a head-end system or estimation processes, or it can be manually created by a user (to override or estimate consumption). Measurement data from these different sources might use different VEE Rules. For example:

- Initial measurements sent a head-end system might use strict VEE Rules
- Initial measurements created by a user (to override or estimate consumption) may use less strict rules
- Initial measurements created by the system to estimate consumption have very few (if any) VEE Rules

You can define different VEE Groups for different roles and sources of incoming measurements. The base package roles are:



Copyright © 2026, Oracle and/or its affiliates.

- **Initial Load**: Used for initial measurement data received from a head-end system or import process
- **Initial Load - Streaming**: Used for initial measurement data received from a streaming import process
- **Manual Override**: Used for initial measurement data manually created by a user or created by the reconciliation process for scalar or subtractive interval measuring component types
- **Estimation**: Used for initial measurement data estimated by the system
- **Projection**: Used for "what if" services that are projecting future energy use for a measuring component
- **Post-ML**: Used for processing rules after machine learning processing, when using Machine Learning Anomaly Scoring.

Refer to About Initial Measurement Data for additional functional information about the reconciliation process.

A measuring component's Measuring Component Type can define "fallback" VEE groups for each of these roles. In addition, an individual measuring component can specific a VEE group for each role. If the measuring component doesn't have a VEE group specified for a role, the "fallback" VEE group defined for the measuring component type is used.

The VEE process has a number of different objects that are detailed below that allow for configurability and flexibility in how the process executes.

### VEE Groups

VEE groups are collections of VEE Rules that are applied to initial measurement data. During the VEE process, the system executes the VEE Rules defined in each VEE group. The rules within a VEE group are defined in a specific sequence, allowing control over the order in which the rules are executed.

VEE groups can be associated to a specific measuring component, or to a measuring component type (or both). VEE groups associated with a measuring component type are applied to all measuring components of that type, while those associated to a specific measuring component are applied only to that measuring component. VEE groups associated to a measuring component override those assigned to a measuring component type.

When using Machine Learning Anomaly Scoring (LINK), you can define a VEE Group that is executed after the machine learning processing ..

For information on configuration, reference the related section for Configuring VEE Groups in the Administration Guide.

### VEE Rules

The specific validation and estimation processing performed on initial measurement data is defined in individual VEE Rules, each performing a specific set of targeted logic. The base product contains many VEE Rules you can use in your implementation, but you can also create your own custom VEE Rules.

Some VEE Rules generate VEE Exceptions if the initial measurement data fails the conditions specified for the rule. Other rules override measurements, changing measurement values as dictated by the rule's parameters. Some rules can both create exceptions and override the measurement as part of a single process. By convention, VEE Rules change the Post-VEE quantities of initial measurement data, but VEE Rules can change anything on an initial measurement.


Copyright © 2026, Oracle and/or its affiliates.

For a list of validation VEE Rules, reference the About IMD Validations section. For a list of estimation VEE Rules, reference the About IMD Estimations section. For a list of general VEE Rules that handle decision-making within the VEE process, reference the Decision-making VEE Rules section.

For information on configuration, reference the related section for Configuring VEE Rules in the Administration Guide.

### VEE Exceptions

Each VEE Rule defines an exception type and severity that specify how exceptions are tracked by the system. When an initial measurement fails a validation, an exception of the type specified for the failed VEE Rule is created. A single initial measurement can have multiple exceptions, one (or more) for each rule the measurement fails. This allows users to see all of the problems detected during the VEE process.

There are three levels of severity for VEE Exceptions:

- **Info**: Used to highlight something interesting, but not sufficient to cause the initial measurement to be put into the Exception state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues.
- **Issue**: Used to report a problem that will prevent the initial measurement from being finalized. Multiple "issue exceptions" can be created during VEE processing. If at least one issue exists after all rules have been applied, the initial measurement is transitioned to the Exception state.
- **Terminate**: Used to report a severe issue that will cause the VEE process to stop and the initial measurement to be transitioned immediately to the Exception state. Only one terminate exception can be issued (as the first one causes VEE to stop on an initial measurement).

Note that exceptions are not deleted when an initial measurement is adjusted or corrected. After any issues are corrected or the initial measurement is overridden (or manually completed), the exceptions persist in a closed state for reporting purposes.

For information on configuration, reference the related section for Configuring Exception Types in the Administration Guide.

### VEE Rule Eligibility Criteria

Each VEE Rule may optionally have eligibility criteria that controls if the rule is applied. This feature can greatly reduce the number of VEE groups you need to create, because it allows a single VEE group to have conditional VEE Rules based on eligibility criteria (rather than requiring a distinct VEE group for every combination of VEE Rules). Below are a few examples of ways you can use eligibility criteria:

- You might create a rule that compares interval consumption against related scalar consumption (such as might be the case with a device with both interval and scalar measuring components). This rule might use eligibility criteria that specifies that the rule is only applied if the initial measurement's measuring component has a corresponding scalar register measuring component.
- You might want to have a common VEE group but only have certain rules execute to execute based on Unit of Measure. An eligibility criteria can be setup so that a VEE Rule only executed for kWh whereas is skipped for kVARh.
- Another example might be a rule that compares the current consumption against standard consumption for measuring components of a certain type for the first six months after installation. You might create eligibility criteria that specifies that the rule is only applied if


Copyright © 2026, Oracle and/or its affiliates.

the measuring component's Device Configuration has been installed at a Service Point for less than six months.

For information on configuration, reference the related section for Configuring VEE Rules in the Administration Guide.

## Decision-Making VEE Rules

In addition to the VEE Rule Eligibility Criteria, there are VEE Rules delivered as part of the base product that help with decision-making when executing VEE (listed below). For more information on how each rule executes and can be configured, follow the link provided on the rule.

| VEE Rule Name | Purpose |
|---|---|
| Exception Handler | This rule allows for termination of the VEE process based on a configurable set of exceptions being present for the IMD. This rule also allows a unique To Do Type to be generated based on a group of exceptions. |
| Execute VEE Group | This rule performs a call to execute a separate VEE Group which includes execution of all VEE Rules within that group. |
| Successful Termination | This rule allows VEE to be successfully terminated based on a list of exceptions. |
| VEE Group Matrix (Factor) | This rule provides a way to choose different instances of a VEE Rule using a Factor. This factor leverages characteristics that are defined on Service Point, Device, or Measuring Component. |

# Validations

- About IMD Validations
- Reviewing and Resolving VEE Exceptions
- Usage Transaction Validations

## About IMD Validations

Validation of the Initial Measurement Data on the way in to Oracle Utilities Meter Data Management occurs by leveraging the diverse library of VEE Rules. Below is a list of the validation VEE Rules that are provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| VEE Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Consecutive Interval Check | Interval | This validation rule flags any combination of consecutive intervals within Initial Measurement Data based on the values of the data or the condition codes of the data. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Duplicate IMD Check | Interval or Scalar | This rule allows for a duplicate IMD to be flagged. |
| Dynamic Comparison Validation | Interval or Scalar | This powerful, flexible validation rule compares measurements to historical statistics for the related Service Point. The system will maintain statistics such as the following: sum, min, max, average, median, zero value count, outage count, and standard deviation. Then you define formulas (no programming required) for the comparison of current measurements to these statistics. |
| Ensure IMD Exists for Sibling MCs | Interval or Scalar | This rule validates that an IMD exists for all of the other measuring components associated to the same Device Configuration as the current measuring component, for the same period of time. |
| Final Measurement Replacement | Interval or Scalar | This validation rule allows you to define a variety of configuration options to decide if new data should replace existing measurements. The options include value change thresholds, percentage change thresholds, as well as condition code ranking. |
| High/Low Check | Interval or Scalar | This validation rule compares the total consumption of the current IMD to historical values. The comparison is normalized based on average daily usage (ADU). If the current IMD is too high or too low compared to historical data then an exception is thrown. |
| Inactive Measurement Check | Interval or Scalar | This validation rule flags any Initial Measurement Data received on a device that is either disconnected, uninstalled, and/or not connected to a Usage Subscription. |
| Interval Size Validation | Interval | This rule validates that the interval size (in seconds) supplied with the Initial Measurement is equal to the interval size defined on measuring component's type. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Interval Spike Check | Interval | This rule checks for spikes within an IMD and generates an exception if one is found. |
| Multiplier Check | Interval or Scalar | This rule validates that the register multiplier supplied with the IMD is equal to the multiplier stored on the measuring component. If not, an exception is created using the register multiplier exception type and severity configured on the rule. |
| Negative Consumption Check | Interval or Scalar | This rule flags any IMD where the total consumption is negative. |
| Prolonged Estimation Check | Interval or Scalar | This validation rule creates an alert when a device has been estimated for an extended period of time. |
| Raise Missing Quantity Exception | Interval | This rule flags any missing interval data. |
| Sum Check | Interval or Scalar | This rule is used to compare the difference between interval data to scalar data for a period of time, or between a set of TOU scalar reads to a "Sum" scalar reading. |
| Unit of Measure Check | Interval or Scalar | This rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type. |
| Zero Consumption Check | Interval or Scalar | This rule checks if the total consumption for the IMD is zero. There is also a check for whether an outage occurred during the same time as the zero consumption to provide ways to avoid exceptions in that case. |

Refer to About VEE for additional functional information.

# Reviewing and Resolving VEE Exceptions

All To Dos generated by a VEE Exception will link back to the Initial Measurement Data that originally failed to create the VEE Exception.

Users have a number of options for dealing with initial measurements with exceptions.

- After correcting the cause of the issues that triggered the exceptions, a user can reinitiate the VEE process for the IMD using the **Perform VEE** button.
- A user can **Discard** the initial measurement.


Copyright © 2026, Oracle and/or its affiliates.

- A user can use the **Force Complete** button on the IMD in the scenario that they want to approve the exception and the measurement information in the IMD. *Note:* No VEE processing is performed on initial measurement data that is force completed. Regardless of the action taken by the user, the system will complete any open To Do Entries that created when the initial measurement entered the Exception state.

Refer to About VEE for additional functional information.

## Usage Transaction Validations

There are also validations available during the execution of usage calculation. For more information on these reference the Reviewing and Resolving Usage Transactions section.

# Estimations

- About IMD Estimations
- Usage Transaction Estimations

## About IMD Estimations

Below is a list of the estimation VEE Rules provided as part of base product that are responsible for filling the missing data within an IMD. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| VEE Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Interval Adjustment From Scalar | Interval | This rule performs adjustments to interval values based on the values from the associated scalar data on the same device. |
| Interval Averaging Estimation | Interval | This rule finds historical interval data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD. |
| Interval Create Estimation IMD for Gap | Interval | This estimation rule creates a new Estimation IMD to fill gaps between the interval data received from the Head End.<br>***Note:*** *this estimation rule acts differently from the other rules as it creates a new IMD rather than filling in missing values in an estimation IMD created from periodic estimations.* |
| Interval Interpolation Estimation | Interval | This rule attempts to interpolate gaps within an IMD using prior and subsequent intervals as starting points for linear interpolation. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Interval Profile Estimation | Interval | This rule estimates any missing interval data for the IMD based on a referenced Profile Factor. |
| Interval Proxy Day Estimation | Interval | This rule checks for days with similar temperature and uses the measurement data from those days as the basis of interval estimation for the in-flight initial measurement. |
| Scalar Calculation From Interval | Scalar | This rule performs adjustments to scalar values based on the values from the associated interval data on the same device. |
| Scalar Estimation | Scalar | This rule finds historical scalar data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD. |
| Scalar Profile Estimation | Scalar | This rule will estimate any missing scalar data for the IMD based on a referenced Profile Factor. |
| Scalar Proration | Scalar | This rule prorates the value of a scalar reading that has two valid scalar readings on either side as boundaries. It will also take into account any related interval data within the same period to exclude from the calculation. |
| Subtractive Interval Adjustment Rule | Subtractive Interval | This rule performs adjustments to qualifying interval consumption values for subtractive interval measuring components based on an adjustment target calculated using a start and stop reading for the period that encapsulates the intervals to adjust. |

Refer to About VEE for additional functional information.

## Usage Transaction Estimations

There are also estimations available at the time of usage calculation. For more information on these reference the Reviewing and Resolving Usage Transactions section.

# Editing

- About IMD Editing


Copyright © 2026, Oracle and/or its affiliates.

- Editing Initial Measurements
- Generating New Initial Measurements

## About IMD Editing

At times, the values received in as Initial Measurement Data need to be edited by a user. This is covered in the Editing Initial Measurements section.

If measurement data is found to be incorrect, a new IMD must be generated to correct the data. This is covered in the Generating new Initial Measurements section.

Refer to About VEE for additional functional information.

## Editing Initial Measurements

Initial measurements can be manually edited via functions available in the Interval Initial Measurement Lens or Scalar Initial Measurement zones in the Initial Measurement portal. Manual edits can include changes to the measurement values or condition codes.

By default, manual edits made by users of these zones are not logged on the Log tab.

Refer to Configuring Measurement Logging for additional information on how initial measurement changes can be logged.

## Generating New Initial Measurements

The main location for viewing measurements and generating new IMDs for any discrepancies is through the 360 Degree View Measuring Component portal. On the portal many of the zones provide ways to apply measurement functions that will generate a new IMD.

# 360 Degree and Unified Search

- Using 360 Degree Search
- Using Unified Search

## Using 360 Degree Search

The 360 Degree Search feature allows users to search for devices, measuring components, service points, and/or usage subscriptions using a variety of search criteria.

From the search results list, you can navigate to the 360 Degree View portal to view data on one of the following tabs:

- Measuring Component
- Device
- Service point
- Usage Subscription
- Contact


Copyright © 2026, Oracle and/or its affiliates.

When accessed, the 360 Degree View opens on the tab corresponding to the type of object (device, measuring component, etc.) you select. For example, if you click the Measuring Component description, the Measuring Component tab opens.

Refer to 360 Degree Views for more details about 360 Degree View.

The following options are available for 360 Degree Search:

- **Search by Name:** Allows search by name/address OR contact identifier OR usage subscription identifier. To search by address, click Go To Search by Address. To search by device, click Device.
- **Search by Address:** Allows search by address/name OR service point identifier. To search by name, click Go To Search by Name. To search by device, click Device.

> **Note**
>
> "Search by Address" search results contain service points with one or more active usage subscriptions only.

- **Search by Device:** Allows search by device identifier OR measuring component identifier. To search by name, click Go To Search by Name. To search by address, click Address.

# Using Unified Search

The Unified Search feature allows you use the toolbar search field to search for business entities as well as search for menu items. See Unified Search in the User Interface Standard Features section for more information about the general use of the unified search field.

In Oracle Utilities Meter Data Management and Oracle Utilities Meter Solution Cloud Service, the Unified Search field allows searching by a service point's address (Address), a usage subscription's contact (Name), or a meter’s badge number (Badge Number) or serial number (Serial Number).

You can use the following filters to indicate your specific search criteria:

| Filter | Filter Rule |
|---|---|
| Address | One or more numeric characters followed by space and an alphanumeric character |
| Name | Minimum of 5 characters excluding whitespaces |
| Badge Number | Minimum of 5 characters excluding whitespaces |
| Serial Number | Minimum of 5 characters excluding whitespaces |

For example, to search for a service point located at 1401 Flower Street, you could enter “1401 Flower”.

To search explicitly using a specific filter, precede the filter value with one of the following predefined hints:

| Filter | Hint | Example |
|---|---|---|
| Address | ad: | ad:1401 Flower |
| Name | n: | n:Smith, J |


Copyright © 2026, Oracle and/or its affiliates.

| Filter | Hint | Example |
|---|---|---|
| Badge Number | bn: | bn:12345 |
| Serial Number | sn: | sn:6789 |

Examples:

- To search for a service point located at 1401 Flower Street, you could enter "ad:1401 Flower" (hint) or “1401 Flower” (no hint)
- To search for a person named John Smith, you could enter "n:Smith, J" (hint) or “Smith,J” (no hint)
- To search for a meter whose badge number is 12345, you could enter "bn:12345" (hint) or “12345” (no hint)
- To search for a meter whose serial number is 6789, you could enter "sn:6789" (hint) or “6789” (no hint)

Clicking the **Advanced Search** link opens the **360 Search by Device** portal and the **Search by Device** zone.

Fore information about configuring the Unified Search for your implementation, refer to Understanding Unified Search in the *Administrative User Guide*.

### Search Results

Each result in the search result list may include any of the following:

- Contact Info (highlighted when using the Name filter) or “No Contact Information”
- Service Point Info (highlighted when using the Address filter) or “No Service Point Information”
- Measuring Component Info (highlighted when using the Badge Number or Serial Number filter)
- Filter Name and Search Criteria (always displayed)

To select an item from the search results, click the result.

# 360 Degree Views


- 360 View Measuring Component
- 360 Degree View Device
- 360 Degree View Service Point
- 360 Degree View Usage Subscription
- 360 Degree View Contact


## 360 View Measuring Component


- Using 360 Degree View Measuring Component
- Applying Measurement Functions




Copyright © 2026, Oracle and/or its affiliates.

# Using 360 Degree View Measuring Component

This portal is used to view a broad set of information related to a Measuring Component.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Measuring Component has been selected you are brought to the 360 Degree View Measuring Component portal.

The following zones may appear as part of the portal's **Measuring Component** tab page:

- **Measuring Component Overview**: Displays the Service Point, device, Device Configuration, Install Event, and Usage Subscription for the selected device.
- **Device Configuration Overview**: Displays an overview of the measuring component's parent device's configurations over time, including the measuring components associated with each configuration, and Service Points (along with the relevant Install Events) at which the device's configurations have been installed over time, if any exist.
  - To create a new reading for a scalar measuring component associated to a Device Configuration, click the **New Reading** link in the zone title bar.
- **Interval - Initial Measurement Overlay**: Overlays initial interval measurements (either pre-VEE or post-VEE) against the final measurements for a given period in graphic and tabular formats.
- **Final Values Overlay**: Graphs final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components. The zone also permits overlaying data from the same measuring component for different time periods, as well as data from measuring components that measuring different quantities, such as temperature.
- **Interval - Audit View**: Allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the curve and see how the measurements looked at different points in time.
- **Time of Use By Day**: Displays daily TOU-mapped usage data for an interval measuring component based on a user- defined time period and TOU map.
- **Time of Use Overlay**: Displays an overlay of the TOU periods on a final interval measurement along with totalized TOU consumption based on a user-defined time period and TOU map.
- **Scalar - Initial Measurement Overlay**: Overlays initial scalar measurements (either pre-VEE or post-VEE) against the final measurements for a given period in graphic and tabular formats.
- **Scalar - Audit View**: Allows users to view a scalar measurement for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the measurement and see how the measurements looked at different points in time.
- **Measurements**: Displays measurement details for the current measuring component in a tabular format. (split into two zones for v2.2)
- **Usage Subscriptions Directly Linked to MC**: Displays Usage Subscriptions that are directly linked to current measuring component.

Functions are available in many of these zones that allow you to interact with the measurement data and/or create new data. For more information of these functions refer to the Applying Measurement Functions section.


Copyright © 2026, Oracle and/or its affiliates.

## Applying Measurement Functions

Many of the Interval measurement zones on the Measuring Component tab of the 360 Degree View provide access to the following functions:

- **Create Scratchpad**: Creates a scratchpad measuring component along with new initial measurement data by copying all or some of the final measurements from the measuring component displayed in the zone.
- **Create / Override**: Creates new initial measurement data for a selected measuring component for all or part of a selected time period. This function can either copy existing final measurements or create new measurement data for the initial measurement it creates.
- **Export to Excel**: Exports measurement data to a comma-separated values spreadsheet.
- **New Reading**: Creates a new interval initial measurement reading for the measuring component.
- **Rederive Values**: Rederives the measurement's values displayed in the zone.
- **Save As**: Creates new initial measurement data for a measuring component other than that being displayed by copying all or some of the final measurements shown in the zone.

Many of the Scalar measurement zones on the Measuring Component tab of the 360 Degree View provide access to the following functions:

- **Convert to Interval**: Converts scalar consumption into interval consumption. The converted consumption is held in a new scratchpad measuring component.
- **Create / Override**: Creates new initial measurement data for a selected measuring component for all or part of a selected time period. This function can either copy existing final measurements or create new measurement data into the initial measurement it creates
- **Export to Excel**: Exports measurement data to a comma-separated values spreadsheet.
- **New Reading**: Creates a new scalar initial measurement reading for the measuring component.
- **Rederive Scalar Values**: Rederives the measurement's values displayed in the zone.

# 360 Degree View Device


- Using 360 Degree View Device


## Using 360 Degree View Device

This portal is used to view a broad set of information related to a Device.

You can access the portal by selecting **Main**, then **360 Search**, then [options]. You are provided with the option of searching by Name, Address, or Device. Once a Device has been selected you are brought to the 360 Degree View Device portal.

The following zones may appear as part of the portal's **Device** tab page:

- **Device Overview**: Displays the device, Service Point, Install Event, and Usage Subscription for the selected device.



Copyright © 2026, Oracle and/or its affiliates.

- **Device Configuration Overview**: Displays an overview of the device's configurations over time, including the measuring components associated with each configuration, and Service Points (along with the relevant Install Events) at which the device's configurations have been installed.
- **Scalar Readings**: Displays a list of scalar readings for scalar measuring components associated with the current device and Device Configuration.
  - To create a new reading for scalar measuring components associated with the Device Configuration, click **New Reading**.
  - To edit an existing reading, click the edit icon for the reading you wish to edit. Editing a reading creates new initial measurements that are validated. If these edits do not pass validations, including VEE Rules, the changes are not saved.

# 360 Degree View Service Point


- Using 360 Degree View Service Point


## Using 360 Degree View Service Point

This portal is used to view a broad set of information related to a Service Point.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Service Point has been selected you are brought to the 360 Degree View Service Point portal.

The following zones may appear as part of the portal's **Service Point** tab page:

- **Service Point Overview**: Displays the Service Point, device, Device Configuration, Install Event, and Usage Subscription for the selected Service Point.
- **Measuring Component Types**: Lists the measuring component types configured for the Service Point. To display consumption history for a measuring component, click the **Broadcast** icon next to the Measuring Component Type. Doing so will display either the Interval Consumption History or the Scalar Consumption History.
- **Interval Consumption History**: Displays a graph of interval consumption for a broadcast interval measuring component type.
- **Scalar Consumption History**: Displays a graph of scalar consumption for a broadcast scalar measuring component type.
- **Timeline**: Displays a timeline of events for the Service Point, including usage transactions, activities, and VEE exceptions. Each event is depicted as a button containing the date on which the event occurred. To search for a specific month and year in the Timeline zone, select the month and year from the drop-down lists and click the **Search** icon. To navigate forward and back along the timeline, use the left and right arrows.
- **Service Issue Monitors Related to SP**: displays a list of service issue monitors and related service investigative orders related to the current Service Point, in reverse chronological order.

# 360 Degree View Usage Subscription


- Using 360 Degree View Usage Subscription




Copyright © 2026, Oracle and/or its affiliates.

## Using 360 Degree View Usage Subscription

This portal is used to view a broad set of information related to a Usage Subscription.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Usage Subscription has been selected you are brought to the 360 Degree View Usage Subscription portal.

The following zones may appear as part of the portal's **Usage Subscription** tab page:

- **Usage Subscription Overview**: Displays the Usage Subscription, main customer, Service Point, and direct channels for the selected Usage Subscription.
- **Usage Transaction History**: Lists usage transaction records for the Usage Subscription, including the start date/time, end date/time, description, and duration (in days) for each.
- **Timeline**: Displays a timeline of events for the Usage Subscription, including usage transactions, activities, and VEE exceptions. Each event is depicted as a button containing the date on which the event occurred. To search for a specific month and year in the Timeline zone, select the month and year from the drop-down lists and click the **Search** icon. To navigate forward and back along the timeline, use the left and right arrows.

# 360 Degree View Contact


- Using 360 Degree View Contact


## Using 360 Degree View Contact

This portal is used to view a broad set of information related to a Contact.

You can access the portal from the **Main > 360 Search > [options]**. You are provided with the option of searching by Name, Address, or Device. Once a Contact has been selected you are brought to the 360 Degree View Contact portal.

The following zones may appear as part of the portal's **Contact** tab page:

- **Contact Overview**: Displays contact information (name, phone number, email address, etc.) for the selected contact.
- **Contact's Service Points**: Lists the Service Points related to the contact, including the relationship type (main contact, main customer, etc.) for each.
- **Contact's Usage Subscriptions**: Lists the Usage Subscriptions related to the contact, including the relationship type (main contact, main customer, etc.) for each.

# Usage


- Introduction to Usage Calculation
- Usage Subscriptions
- Usage Subscription Market Participants
- Usage Transactions




Copyright © 2026, Oracle and/or its affiliates.

- Usage Calculation Requests
- Usage Subscription Quantities
- Time of Use Maps
- Dynamic Options
- Consumption Extract Requests
- Contacts


# Introduction to Usage Calculation


- About Usage Calculation


## About Usage Calculation

Oracle Utilities Meter Data Management can calculate and publish usage calculated from measurement data to service providers on an ongoing basis. In addition, external systems can request usage whenever needed. Usage calculations derive a usage transaction's usage quantities using the measurements linked to a usage subscription's service points.

Before bill determinants can be calculated, you must first create a Usage Subscription. The usage subscription defines the external system to which the data should be sent as well as the usage calculation group that will be used for calculation. The usage calculation rules defined for the usage calculation group are responsible for performing the specific calculations, validations, or estimations for bill determinants.

Usage Service Quantities (bill determinants) are derived from the Measurements of the measuring components installed at the Usage Subscription's Service Points during the calculation period. A Service Point is linked to measuring components through an Install Event linked to the measuring components' Device Configuration.

The results of all calculations performed by Usage Subscriptions are stored as a Usage Transaction.

At any instance in time:

- A Usage Subscriptions may be linked to multiple Service Points
- A Service Point may be linked to a single Device Configuration
- A Device Configuration may have multiple measuring components

The calculation period for bill determinant calculations can span many days and over this period:

- The Service Points linked to the Usage Subscription can change (Service Points can be added and removed)
- The Device Configurations installed at the Service Point can change (due to device reconfigurations and meter exchanges)

This means that values for each Usage Subscription can be calculated using multiple Service Points and measuring components.

The Usage Calculation process has a number of different objects that are detailed below that allow for configurability and flexibility in how the process executes.



Copyright © 2026, Oracle and/or its affiliates.

### Usage Calculation Groups

Usage Calculation Groups are collections of usage calculation rules that are used to either calculate bill determinants or validate bill determinants. During the Usage Transaction process, the system executes the usage calculation rules defined in the usage calculation group referenced on the usage subscription. The rules within a usage calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

For information on configuration, reference the related section for Configuring Usage Calculation Groups in the Administrattive User Guide.

### Usage Calculation Rules

The specific calculation, validation, or estimation processing performed on a Usage Transaction is defined in individual usage calculation rules, each performing a specific set of targeted logic. The base product contains many usage calculation rules you can use in your implementation, but you can also create your own custom usage calculation rules.

For a list of validation usage calculation rules, reference the About Usage Validations section. For a list of estimation usage calculation rules, reference the About Usage Estimations section. For a list of all other usage calculation rules that handle calculation of bill determinants, reference the Inventory of Other Usage Calculation Rules section.

For information on configuration, reference the related section for Configuring Usage Calculation Rules in the Administration Guide.

### Usage Transaction Exceptions

Each usage calculation rule defines an exception type and severity that specify how exceptions are tracked by the system. When a Usage Transaction fails some part of a usage calculation rule, an exception of the type specified for the failed usage calculation rule is created. A single Usage Transaction can have multiple exceptions, one (or more) for each rule that failed. This allows users to see all of the problems detected during the Usage Calculation process.

There are three levels of severity for Usage Transaction Exceptions:

- **Information**: Used to highlight minor issues, but not sufficient to cause the Usage Transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the usage transaction from being sent. Multiple "issue exceptions" can be created during usage transaction processing. If at least one issue exists after all rules have been applied, the usage transaction is transitioned to a failure state requiring review and approval.
- **Terminate**: Used to report a severe issue that will cause the Usage Calculation process to stop and the Usage Transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop on for a Usage Transaction). This should be used for cases where manual override / approval isn't accurate. For example, a "Curve Not Continuous" error that says the interval data doesn't cover the full usage period should always be set to Terminate as an Exception Severity.

Note that exceptions are not deleted when a Usage Transaction is adjusted or corrected. After any issues are corrected or the Usage Transaction is overridden (or manually completed), the exceptions persist in a closed state for reporting purposes.

In addition to exceptions, usage processing can also trigger the creation of To Do Entries related to failed validations. If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the usage transaction is transitioned to the Exception state. The


Copyright © 2026, Oracle and/or its affiliates.

To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the usage transaction.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

For information on configuration, reference the related section for Configuring Usage Transaction Exception Types in the Administration Guide.

**Usage Calculation Rule Eligibility Criteria**

Usage calculation rule eligibility criteria are user-definable conditions that could cause a given usage calculation rule to be applied or skipped. This can involve the evaluation of some attribute of the usage subscription or service point, or something else entirely. A usage calculation rule can have multiple eligibility criteria for determining if the rule should be applied or skipped, based on a user-defined sequence.

For information on configuration, reference the related section for Configuring Usage Calculation Rules in the Administration Guide.

# Usage Subscriptions


- About Usage Subscriptions
- Maintaining Usage Subscriptions


## About Usage Subscriptions

A Usage Subscription is an ongoing request to send one or more Service Points' usage to one or more external systems. The usage subscription defines which usage calculation group should be used to calculate service quantities (often referred to as bill determinants).

Usage subscriptions must reference a:

- Service Provider. The service provider is used as the identity of the subscribing system. In other words, you must set up a service provider for any system that subscribes to bill determinants.
- Contact: Contacts are individuals or business entities with which a company has contact. A contact exists for every individual or business related to a usage subscription. A single usage subscription can have many contacts, and a single contact may be referenced on many different usage subscriptions. Contacts have a 1-to-1 correlation with a "person" in a customer information system (CIS) and the CIS is considered the system of record for contact information.

**An Aside: No Account Object Exists**

Oracle Utilities Meter Data Management is not considered the system of record for accounts or Usage Subscriptions. The customer information system (or some other system) is considered the system of record for this type of information. In order to minimize the amount of data that must be synchronized between systems, account and service agreement oriented attributes used by the meter data products are held on Usage Subscriptions. For example, if an account's ID and its customer class are relevant to usage calculations, each Usage Subscription must reference both elements. This is an important distinction to keep in mind when creating custom Usage Subscriptions for your implementation.

Refer to About Usage Calculation for a broader functional view of the overall process.



Copyright © 2026, Oracle and/or its affiliates.

## Maintaining Usage Subscriptions

This portal is used to display and maintain a Usage Subscription.

Refer to About Usage Subscriptions or About Usage Calculation for additional functional information.

You can access the portal from **Menu > Usage > Usage Subscription**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription**: Defines basic information about the usage subscription, including factor overrides, usage calculation groups, and service points.
- **Usage Subscription Market Participants**: Lists market participants associated with the usage subscription (via Usage Subscription Market Participants)
- **Sub Usage Subscriptions**: Lists the first fifty "sub" usage subscriptions for the master usage subscription displayed in the Usage Subscription zone. This zone is only displayed for "master" usage subscriptions.

# Usage Subscription Market Participants


- About Usage Subscription Market Participants
- Maintaining Usage Subscription Market Participants


## About Usage Subscription Market Participants

Usage subscription market participants connects a Market, Market Participant and Usage Subscription together over a specified date range. Usage subscription market participants can be created either via the **Usage Subscription Market Participant** portal, or by adding a market participant to a usage subscription via the **Usage Subscription** portal.

Usage subscription market participants are defined by the following:

- **Usage Subscription**: The usage subscription for which this usage subscription market participant was created.
- **Market Relationship Type**: The market relationship (Distributor, Supplier, etc.) of the market participant and its role within a market.
- **Start Date/Time**: The start date/time for when the record is applicable.
- **End Date/Time**: The end date/time for when the record is applicable.
- **Market Participant**: The market participant associated with the **Usage Subscription**, including an optional ID used by the market participant to uniquely identify the account (usage subscription).

## Maintaining Usage Subscription Market Participants



Copyright © 2026, Oracle and/or its affiliates.

You use the **Usage Subscription Market Participant** portal is used to display and maintain usage subscription market participants.

Refer to About Usage Subscription Market Participants for additional functional information.

You can access the portal from the **Main > Usage > Usage Subscription Market Participant**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Market Participant**: This zone displays the basic attributes of the usage subscription market participant
- **Sub Usage Subscriptions**: lists sub usage subscriptions for the usage subscription market participant's usage subscription

# Usage Transactions


- About Usage Transactions
- About Usage Validations
- Reviewing and Resolving Usage Transactions
- Triggering Adjustment Notifications
- Inventory of Other Usage Calculation Rules


## About Usage Transactions

Usage Transactions are records of bill determinant calculations for a Usage Subscription. All of the Service Quantities calculated are stored as part of the Usage Transaction.

Most requests for usage result in the creation of a usage transaction, but it is possible for an external application to invoke the usage calculation engine real-time. In other words, usage can be retrieved for a usage subscription real-time without creating a usage transaction. This technique is only recommended for online requests, not as part of batch processes.

### Usage Calculation Group Determination

The first step in calculating the usage is to determine the usage calculation group holding the rules set used in calculating or validating the bill determinants. There are two ways of configuring the usage calculation group to use in the processing. It may be specified on the usage subscription during its creation with an effective and expiration date. If there are no usage calculation groups in effect for the usage transaction, the fallback usage calculation group specified in the usage subscription type will be used. The other way is to specify an algorithm – Determine Usage Calculation Group Override, on the usage subscription type. This option provides flexibility in cases where special or custom logic will be required to identify the usage calculation group. Base product delivered an algorithm that identifies the usage calculation group based on the rate and installed device. Additional details can be found in the Detailed Description of the D2-DRVUSGGRP Algorithm Type.

### Calculation Period

Usage transaction requests must specify the date range for the usage transaction. This date range is referred to as the calculation period for the usage transaction. The dates that define



Copyright © 2026, Oracle and/or its affiliates.

the calculation period are specified by a subscribing system when it requests a usage transaction.

Some usage calculation rules have been enhanced to support usage calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing measurement data and usage transactions. When processing usage calculations for measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the measuring component type.

### Multiple Device Configurations and the Calculation Period

In situations where a meter exchange has taken place at a service point, more than one device configuration can be in effect during the usage transactions cutoff period (the time period between the **End Date/Time From** and **End Date/Time To** values).

To account for this, the **Meter Exchange Option** on the Usage Subscription Type can be used to specify how the usage calculation should handle multiple device configurations in effect during the cutoff period. If the flag is set to 'Defer Calculation', new device configurations are excluded in the current usage transaction. If the flag is blank or set to 'Calculate Usage', new device configurations are included in the current usage transaction.

### Date Breaks

As noted above, date breaks are used to break up a usage period into sub-periods based on the dates on which rate changes took place for the service point. For example, suppose a subscribing system requests usage for the month of January. The customer for this request has an interval meter, and the customer's usage is calculated by applying a TOU map to their interval consumption. The subscribing system detects that the customer's rate changed in the middle of January (on January 16) and wants the TOU consumption calculated in two "chunks" (before and after the rate change). Because the customer has an interval meter, the exact consumption amounts before and after the rate change can be precisely calculated (as opposed to calculating each period's amount by dividing the total usage by the number of days in each period).

One approach to this situation would be for the subscribing system to request two usage transactions (where each has the desired date range). Another approach is for the subscribing system to request a single usage transaction with date breaks that define the date ranges before and after the rate change. To continue the above example, a usage transaction could be created with a date break on January 16.

If a usage transaction has date break(s), the usage calculation engine segregates the usage into multiple usage periods based on the date breaks. If there are no date breaks in the usage transaction, a single usage period is created for the entire calculation period.

As of the v2.0.0 release, the only way a usage transaction can have date breaks is if the subscribing system supplies these when it requests usage (only the subscribing system knows if and when its prices and pricing rules change during the billing period). If an implementation has additional criteria that causes date breaks, these criteria can be easily added to the usage transaction's business rules.

### Service Quantities

Every usage period created for a usage transaction contains one or more service quantities. Service quantities are calculated by the usage calculation group's rules specified for the usage subscription from which the usage transaction is created. For example, a usage transaction might have service quantities calculated by a single rule that applies a TOU map to the kWh channel on the device configuration installed at the service point.


Copyright © 2026, Oracle and/or its affiliates.

Each service quantity lists details about that quantity, including a UOM (or TOU or SQI as appropriate) and a quantity. In addition, each service quantity also references the source (measuring component) of the quantities (for audit purposes).

### Service Quantity Rounding

The manner in which service quantities are rounded can be defined using the **SQ Rounding Details** parameters. These parameters allow configuration of the method by which quanitities are rounded (Up, Down, or Nearest) and the number of decimal positions retained after rounding.

### Quality Assessment for Bill Print Options

Usage transaction service quantities can also contain information about the quality of the measurements used in their calculations and other information that can be sent to an external customer information or bill print system. This information is included in both the usage transaction and its related outbound message.

### Quality Assessment

The Daily Scalar Usage Calculation Rule, Get Interval Data, Get Scalar Details, and Get Time of Use Mapped Usage usage calculation rules include a "Perform Measurement Quality Assessment" option that indicates (yes or no) if the measurement conditions of the measurements used when calculating usage should be evaluated and included in the resulting usage transaction and outbound message. When enabled (set to "Yes"), the usage calculation rule algorithm evaluates the condition codes of the measurements used in the calculation to determine if they fall within a "Regular" range of conditions (this range is defined by the "Bottom Regular Condition Range" and "Top Regular Condition Range" algorithm parameters). The result of this assessment is displayed in the "Data Quality Assessment" column in the Usage Periods section which indicates if the service quantity is based on "regular" or "non-regular" data. (If the "Perform Measurement Quality Assessment" option is disabled, the "Data Quality Assessment" column displays "No Assessment Done). In addition, the "Source Measurement Quality List" column contains XML that contains a list of measurements aggregated by condition/quality.

### Maximum/Minimum Dates and Times

The Get Interval Data, Get Time of Use Mapped Usage, and Vector and Service Quantity Math usage calculation rules can be configured to identify the date/time of the maximum or minimum measurement value used when calculating the service quantity. For these functions, if "Max" or "Min" is selected as the "Calculation Function" (or the "Comax" and "Comin" functions for the "Vector and Service Quantity Math" rule), the date/time of the maximum/minimum value will be included in the resulting usage transaction and outbound message. The "SQ Highlight Date/Time List" column contains XML that contains a list of maximum/minimum date/times, based on condition/quality.

### Usage Transaction Export Configuration

Usage transactions can also include additional information about its related meters and service points, as well as an indication if the service quantity is based on estimated usage. The "Usage Transaction Export Configuration" master configuration is used to control the export of this data. It holds the device and service point business objects used to supply meter and service point information included in the usage transaction and outbound message.

The "Usage Transaction Export Configuration" master configuration also defines the threshold percentage above which a service quantity being exported via the usage transaction outbound message is marked as an estimate. For example, of the "Estimation Threshold Percentage" is


Copyright © 2026, Oracle and/or its affiliates.

set to 35%, any service quantity being exported that was calculated from measurements containing more than 35% estimated usage is marked "estimated".

### Service Quantities for Scalar Usage

Usage transactions and service quantities for scalar usage differ in some ways from those created from interval usage. For example, suppose a subscribing system requests usage for the month of January for a customer with a scalar meter that has been exchanged mid-month. In this case, the customer's usage is calculated by finding the scalar readings in the requested period (including all meter exchanges), and the subscribing system requires a record of all scalar readings AND a total of their consumption

Scalar rules cannot use date breaks because the system doesn't store interval values so it cannot accurately compute the amount in each period. In this case, scalar usage calculation rules retrieve scalar readings for the service points linked to the usage subscription, and then creates a single usage period for the usage transaction's entire calculation period. The individual scalar readings are captured as "scalar details" in the usage transaction.

To continue the example above, there would be 2 entries in the Scalar Details, one for each meter, and a single usage period that contains the total consumption for the entire calculation period. The tables below illustrate what this might look like.

### Usage Estimation

When calculating usage for monthly read scalar meter, the end read may not be available by the end of the bill cycle window. If the request indicated that estimated measurement is allowed, the Get Scalar Details usage calculation rule creates an estimated end read that will be used to calculate the scalar usage.

### Calculating Usage for Items

Usage transactions can also calculate usage for items installed at service points. Usage for items is calculated through use of Get Item Counts and Consumption usage calculation rule. This rules calculate usage based on the number of items installed at each service point and the average daily service quantity defined for each item's type. The specifics of how usage is calculated for items is based on the service point category (defined on the service point type) at which the items are installed (item or multi-item).

### Hypothetical Calculations - Rate Comparisons

When Oracle Utilities Meter Data Management is integrated with Oracle Utilities Customer Self Service, customers can perform rate comparison calculations to see the impact to their usage and billing charges based on a set of user-defined usage adjustments. Rate comparison usage transactions are flagged "Hypothetical Calculation" in the **Calculation Mode** field.

#### *Profile Accumulation Usage Calculation Rules and Usage Adjustment Profile Factors*

Rate comparison usage transactions can be created through the use of Profile Accumulation usage calculation rule. This rule combine a service point's measurements with "usage adjustment" measurements derived from measurement data for profile measuring components defined as characteristic values for one or more profile factors.

Usage adjustment profile factors are associated to usage adjustment types defined in the Usage Adjustment Type extendable lookup and rates defined in the CC&B Rate Schedule extendable lookup (this association is defined in the Self-Service Master Configuration). When the usage transaction is created, the rate determines the usage calculation group to execute and usage adjustment profile factors to include in the calculation based on the CC&B Rate Schedule extendable lookup.


Copyright © 2026, Oracle and/or its affiliates.

**Note**: The Profile Accumulation usage calculation rule should be configured with eligibility criteria such that they are only executed when the Calculation Mode for the usage transaction is set to "Hypothetical Calculation" (D2HC).

### Including Summary Service Quantities and Interval Data Snapshots with Usage Transactions

Usage transactions can be configured to include summary service quantities and an interval data "snapshot" along with service quantities and other information sent to external systems. This used when sending usage to a billing system that performs interval-based pricing calculations.

Only usage transactions created (in whole or in part) by either the Get Interval Data or Vector and Service Quantity Math usage calculation rules can include interval data snapshot information. These rules use the **Extract Interval Data** flag to indicate ("Yes" or "No") if a "snapshot" of the interval data resulting from the usage calculation rule should be sent to external systems when usage from the usage transaction is sent to external systems.

In the base package, the Usage Transaction Outbound Message (D2-UsageTranOutboundMesg) business object is used to create outbound messages containing usage transactions. This business object includes the "Build Summary SQs" (D2-BLDSUMSQ) Pre-Processing algorithm which calculates and creates the interval data snapshot and other summary service quantities.

### How Summary Service Quantities and the Interval Data Snapshot is Calculated

As noted above, the "Build Summary SQs" (D2-BLDSUMSQ) algorithm builds the Summary Usage Periods List based on the usage transaction's details. For every usage period within the usage transaction:

An entry is inserted into the summary service quantity list for every distinct UOM/TOU/ SQI combination in the usage period service quantity list.

An entry is inserted into the service point service quantity list for every distinct service point ID/UOM/TOU/SQI combination in the usage period service quantity list.

Interval data for multiple service quantities are combined into a single interval data curve per UOM/TOU/SQI combination. Intervals are summarized and scaled based on a common SPI

The "Build Summary SQs" (D2-BLDSUMSQ) algorithm uses the "Target Interval Size" parameter to define the interval size of the combined curve when one or more curves are combined with different interval sizes. For example, when combining interval data with 15 minutes intervals (SPI 900) and data with 60 minute intervals (SPI 3600), a Target Interval Size of 60 minutes (SPI of 3600) would result in combined data with an interval size of 3600 (the 15 minute intervals would be scaled up to 60 minute intervals before being combined with the 60 minute interval data).

Refer to About Usage Calculation for a broader functional view of the overall process. For information on how to configure the request and response process for Usage Transactions, reference the Integrating Usage Transactions section of the Admin Guide.

## About Usage Validations

For the sake of keeping the volume of To Dos to a manageable level it may make sense to shift certain validations to the usage transaction process rather than the IMD load process. The validation should be done on a usage transaction when the result is only crucial for the systems receiving usage transactions rather than more real time needs such as online presentation to a customer. Below is a list of the usage calculation rules used for validating


Copyright © 2026, Oracle and/or its affiliates.

usage transactions that are provided as part of base product. To get more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Usage High/Low Rule | Interval or Scalar | This validation rule compares the Service Quantities of the Usage Transaction to historical values. If the current value is too high or too low compared to historical data then an exception is thrown. |
| Validate Against Tolerance | Interval or Scalar | This usage calculation rule is used to validate the calculated usage against a tolerance value. The tolerance value may either come from the specified value or tolerance factor defined in the usage calculation rule. |
| Business Flag Hold | Interval or Scalar | This usage calculation rule can stop a usage transaction from proceeding when there have been business flags for the applicable service points. The hold can either be indefinite or set to expire a configurable amount of time prior to the calculation window ending. |

Refer to About Usage Transactions or About Usage Calculation for additional functional information.

# Reviewing and Resolving Usage Transactions

This portal is used to review and/or resolve a Usage Transaction.

Refer to About Usage Transactions or About Usage Calculation for additional functional information.

You can access the portal from the **Main > Usage > Usage Transaction**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Transaction**: Defines basic information about the usage transaction, including usage period date ranges, date breaks within the usage period, scalar usage details, and usage quantity details.
- **Service Quantity Overlay**: Displays interval data for each service quantity for the usage transaction's usage period
- **Time of Use By Day**: Displays TOU-mapped interval data by day for the usage transaction's usage period

A number of important actions are available for a Usage Transaction depending on its status:

| Action | Allowed Statuses | Details |
|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Approve | Approval In Progress<br>Issue Detected (only when the exceptions are marked to require approval) | Allows a user to bypass any exception or failure message for the Usage Transaction and attempt to send it to the requesting system. |
| Re-Calculate | Approval In Progress<br>Issue Detected | Reruns the entire process of calculation along with any validations (all usage calculation rules) for the Usage Transaction. |
| Subsequent Correction | Sent | Allows a user to indicate that the usage transaction was corrected after being sent. Sets the status of the usage transaction to Subsequent Correction. |
| Resend | Send Error | Allows a user to re-send a usage transaction (via its related outbound message) to an external system. Used when an error occurs when sending usage transactions. |

## Triggering Adjustment Notifications

If any new Measurements are received from a device after that period has been calculated in a Usage Transaction then a discrepancy exists between Oracle Utilities Meter Data Management and the external system (often a customer information system or CIS). As a result, Oracle Utilities Meter Data Management provides a method to notify external systems of these adjustments to Usage Transactions. A **Usage Transaction Correction Processor** activity is created for each usage transaction that may have been impacted by an initial measurement. The **Usage Transaction Correction Monitor** batch (D1-UTCRN) is used to process these activities.

On the External Application for CIS, a processing method of "Usage Trans Subsequent Correction Notification" must be configured to specify the message that should be sent to notify CIS when an adjustment occurs for a Usage Transaction in Oracle Utilities Meter Data Management.

The IMD process handles the logic to look for existing Usage Transactions to determine if an adjustment will be required. The Usage Transaction will be moved to a "Subsequent Correction" status once this adjustment occurs.

## Inventory of Other Usage Calculation Rules

Beyond validations and estimations usage calculation rules, the Usage Transaction process involves many other important rules that prepare and calculate Service Quantities. A list of these rules is provided below. For additional detail, follow the provided link that will take you to the Configuration Guide.


Copyright © 2026, Oracle and/or its affiliates.

### Pre-Calculation Usage Calculation Rules

Below is a list of the pre-calculation usage calculation rules provides as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Alignment and Delay Calcluation Rule | Interval or Scalar (Frequently Read) | This rule can be used to handle two main needs:<br>1) aligns all measuring components for a Usage Subscription to the same date (whether on the same device or separate devices)<br>2) delays the usage transaction until the end of the retry window based on the quality of available data and other configured parameters |
| Check Existence of Installed Device | Interval or Scalar | This rule checks for the existence of a device installed on the Usage Subscription's Service Point for the usage period. In the case of multi-items this rule also checks that they are effective for the usage period. |

### Calculation Usage Calculation Rules

The following is a list of the calculation usage calculation rules provided as part of the base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Apply Math (Interval Data) | Interval | This rule is used to perform calculations on interval data and stores the results in the usage transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions).<br>This rule provides aggregated usage for all selected interval measuring components (filter by TOU, SQI & UOM) associated to a usage subscription.<br>This rule can also multiply total usage by a factor using a custom formula. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Daily Scalar Usage Calculation Rul | Scalar | This rule is used to calculate usage of daily scalar measuring components installed in the Service Points associated with a Usage Subscription for the specified usage period. It creates bill determinants by taking the difference between the beginning and ending reading for the bill period.<br>This rule can also be used to provide register readings by measuring component.<br>For consumption values, only the beginning and ending readings are exported<br>This rule supports date breaks (the normal Get Scalar Data rule does not). |
| Get Interval Data | Interval | This rule is used to get interval quantities from interval measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Interval' usage period.<br>This rule retrieves the interval data for measuring components associated to a usage subscription by TOU, SQI and UOM.<br>This rule converts the interval data to another interval length or unit of measure. |
| Get Item Counts and Consumption | Scalar | This rule finds item-based and multi-item-based Service Points linked to the Usage Subscription for the current usage transaction, summarizes the item counts by item type and Service Point, and calculates item-based consumption. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Get Scalar Details | Scalar | This rule is used to get usage from scalar measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Scalar' usage period.<br>This rule creates bill determinants by summing all scalar readings for the bill period. This rule can also be used to provide register readings by measuring component. All readings are exported by this rule.<br>Note: This rule is used for traditional monthly read meters. |
| Get Subtractive Interval Details | Subtractive Interval | This rule is used to get interval quantities from subtractive interval measuring components installed on the service points linked to the usage subscription for the specified 'interval' usage period.<br>It also identifies the start and stop readings for each usage period using subtractive interval readings. |
| Get TOU Mapped Usage | Interval | This rule is used to get time of use quantities from interval measuring components for devices installed at the Service Points linked to the Usage Subscription for the specified 'Interval' usage period. |
| Interval Tier Calculation | Interval | This rule calculates the difference between a source and reference vector.<br>This rule loops through each tier that is configured and calculates the imbalance amount associated to that tier level.<br>This rule breaks down that difference into one to many positive or negative tiers, and create a service quantity for each tier calculated. |
| Profile Accumulation | Interval or Scalar | This rule is used to manipulate a customer's interval data by adding other vectors to it. Those other vectors are derived from a list of profile factors and corresponding characteristic values stored in a list on the usage transaction. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Round and Adjust Usage | Interval or Scalar | This rule copies identified source and target Service Quantities and inserts these as Service Quantities that are rounded and adjusted. |
| Vector and Service Quantity Math | Interval | This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations.<br>Typical uses:<br>Perform math using interval data, e.g., take the difference between two curves, find max values, find coincident peaks, multiply a curve by a value, apply TOU maps, etc.<br>Define complex formulas using various interval curves, profile factor values or calculated service quantities (bill determinant values).<br>Support math functions: sin, cos, square root, etc.<br>Store derived curves in memory that can be used in subsequent calculations<br>Please note, this rule is not as efficient as other rules. |

### Decision-Making Usage Calculation Rules

There are usage calculation rules delivered as part of the base product that help with decision-making when executing running the usage calculation process. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Purpose |
|---|---|
| Execute Usage Calculation Group | This rule performs a call to execute a separate usage calculation group which includes execution of all usage calculation rules within that group. |
| Exception Handler | This rule evaluates the exception list that was accumulated during the execution of the calculation rules against exception criteria configured on this rule. |

# Usage Calculation Requests

- About Usage Calculation Requests
- Maintaining Usage Calculation Requests


Copyright © 2026, Oracle and/or its affiliates.

## About Usage Calculation Requests

Usage calculation request are calculations executed at a customer level where there may be more than one calculation per usage subscription. For example, a customer may have a usage subscription for producing billing determinants but there may be additional calculations that will be executed on a much less frequent basis (annually, quarterly, etc). More specifically, usage calculation requests are used to calculate service point quantities (see About Service Point Quantities for more information).

Usage calculation requests are defined by the following:

- **Usage Calculation Type**: The usage calculation type for the request. Usage calculation types define the service point quantity type and UOM/TOU/SQI for quantities calculated for requests of that type.
- **Start Date/Time**: The start of the date range for the request.
- **End Date/Time**: The end of the date range for the request.

## Maintaining Usage Calculation Requests

You use the **Request** portal to view and maintain usage calculation requests.

Refer to About Usage Calculation Requests for additional functional information.

You can access the portal from the **Main > Usage > Usage Calculation Request**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Request**: defines the basic attributes of the usage calculation request
- **Request Log**: lists log activities for the usage calculation request

# Usage Subscription Quantities


- About Usage Subscription Quantities
- Maintaining Usage Subscription Quantities


## About Usage Subscription Quantities

Usage subscription quantities are quantities that are stored at the usage subscription level. These quantities are calculated at irregular frequencies (such as annually or quarterly) and thus do not make sense as a measured quantity on a device and measuring component. For example, in some deregulated markets the Peak Load Contribution could be stored here and effective dated over time to represent the changing contribution of a given usage subscription to the market wide coincidence peaks.

Usage subscription quantities are defined by the following:

- **Usage Subscription**: The usage subscription for which the quantities were calculated



Copyright © 2026, Oracle and/or its affiliates.

- **Usage Subscription Quantity Type**: The Usage Subscription Quantity Type used for the record
- **Start Date** / **End Date**: The date range for the calculated quantity
- **Usage Subscription Quantities**: Calculated quantities based on **Quantity Identifiers** configured on the Usage Subscription Quantity Type (see Understanding Usage Subscription Quantity Types for more information)

> **Note**
>
> There cannot be more than one Usage Subscription Quantity for the same Usage Subscription and Usage Subscription Quantity Type, for the same time period (defined by the Start Date and End Date parameters).

## Maintaining Usage Subscription Quantities

You use the **Usage Subscription Quantity** portal to view and maintain usage subscription quantities.

Refer to About Usage Subscription Quantities for additional functional information.

You can access the portal by selecting **Main**, then **Usage**, then **Usage Subscription Quantity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Quantity**: This zone displays the basic attributes of the usage subscription quantity

# Time of Use Maps

- About Time-of-use Processing
- About TOU Maps
- Maintaining TOU Maps

## About Time-of-use Processing

A type of rate used at many utilities is known as a "time of use" rate. For these rates, the price is based on when the usage occurs. For example, prices are typically higher during peak consumption periods (commonly referred to as "On Peak" periods) to encourage lower use during these times. The time range when a given price is applicable is known as a time-of-use (TOU) period.

Time of use periods can (and often do) change during the year. For example, in the summer, TOU periods might be defined as follows:

| Time Of Use Period | Time Periods |
|---|---|
| On Peak | 8:00 AM - 9:00 PM Monday - Friday |


Copyright © 2026, Oracle and/or its affiliates.

| | |
|---|---|
| Off Peak | 12:00 AM - 8:00 AM Monday - Friday<br>9:00 PM - 12:00 AM Monday - Friday<br>12:00 AM - 12:00 AM Saturday, Sunday, and Holidays |

While winter TOU periods might be defined as follows:

| Time Of Use Period | Time Periods |
|---|---|
| On Peak | 11:00 AM - 5:00 PM Monday - Friday |
| Off Peak | 12:00 AM - 11:00 AM Monday - Friday<br>5:00 PM - 12:00 AM Monday - Friday<br>12:00 AM - 12:00 AM Saturday, Sunday, and Holidays |

This is an example of a "simple" mass-market time-of-use schedule. Commercial and industrial schedules can be much more complex.

Generally speaking, there are two different types of meters that can be installed at service points that participate in TOU rates:

Scalar meters with dedicated registers for each TOU period. For this type of meter, the scalar measurements are the TOU consumption and no special derivation is necessary. The assumption is that the TOU schedule is programmed into the device and no TOU Map is required within Oracle Utilities Meter Data Management.

Interval meters that measure how much is consumed each interval. For this type of meter, the interval measurements are used to derive the consumption in each TOU period based on the TOU Map defined in Oracle Utilities Meter Data Management.

For information on configuration, reference the related section for TOU Mapping in the Administration Guide.

# About TOU Maps

Interval consumption is mapped into time-of-use periods using a TOU map. TOU maps define the TOU periods for each interval. TOU map data is structured in a format very similar to final measurement data, with specified intervals of time each designated to a TOU period (instead of a measurement value). For example, the table below lists a set of TOU map data and corresponding final measurements for a period of 4 hours on January 1, 2010.

**Table Title**

| TOU Map | Measuring Component | | |
|---|---|---|---|
| **Date/Time** | **TOU Code** | **Date/Time** | **Consumption** |
| 1-Jan-10 3:00pm | ONPEAK | 1-Jan-10 3:00pm | 14.678 |
| 1-Jan-10 4:00pm | ONPEAK | 1-Jan-10 4:00pm | 29.12 |
| 1-Jan-10 5:00pm | ONPEAK | 1-Jan-10 5:00pm | 12.12 |
| 1-Jan-10 6:00pm | OFFPEAK | 1-Jan-10 6:00pm | 17.16 |

Any type of final measurement can be mapped to a TOU period, including derived quantities. While the principal requirement for TOU Maps is to aggregate interval consumption into TOU


Copyright © 2026, Oracle and/or its affiliates.

periods for a Usage Transaction, the system has many zones in the user interface that display interval consumption using TOU periods.

## Maintaining TOU Maps

This portal is used to display and maintain a TOU Map.

Refer to About TOU Maps for additional functional information.

You can access the portal from **Menu > Usage > TOU Map**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Map**: Defines basic information about the TOU map, including dynamic options
- **TOU Map Data List**: Lists TOU map data created from the TOU map

The **Time Of Use Map Data Generation Monitor** batch (D1-TOUTR) is used to generate TOU Map data automatically based on the TOU Configuration. TOU Map data can also be generated manually through the TOU Map Portal in the UI.

# Dynamic Options

- About Dynamic Options
- About Dynamic Option Events
- Maintaining Dynamic Options
- Maintaining Dynamic Option Events

## About Dynamic Options

Dynamic Options specify terms that override how usage is normally calculated, such as a critical peak period that affects the TOU mapping of interval consumption.

There are circumstances and conditions during which the rules for creating TOU map data might need to be calculated differently than according to the utility's standard rules. Examples of this might include critical peak periods, curtailment requests or other demand response events. During these types of events, the TOU rules defined for a TOU map may be overridden. This is done through the use of dynamic options and dynamic option events.

Dynamic options allow flexibility for ad-hoc and special events that might impact how usage calculations are performed. For example, if a utility requests that its customers curtail their usage for a specific time period, a dynamic option and corresponding dynamic option event could be used to represent the curtailment request. During the time period of the curtailment request, usage might be calculated differently (with a different set of TOU periods, for example), which in turn might impact how the customer's bill is calculated.

### Dynamic Options and TOU Maps

To apply a dynamic option (and one or more of its related dynamic option events), you reference the dynamic option on a TOU map, along with a corresponding "dynamic" TOU map to be used during the dynamic option event.


Copyright © 2026, Oracle and/or its affiliates.

For example, with Critical Peak Periods you might add a new TOU period (called "Critical Peak"), and create a new TOU map that is the same as your standard TOU map, but that also includes your Critical Peak TOU period. For example, in the summer your new set of TOU periods might be defined as follows:

| Time Of Use Period | Time Periods |
|---|---|
| Critical Peak | 10:00 AM - 2:00 PM Monday - Friday |
| On Peak | 8:00 AM - 10:00 AM Monday - Friday |
| | 2:00 PM - 9:00 PM Monday - Friday |
| Off Peak | 12:00 AM - 8:00 AM Monday - Friday |
| | 9:00 PM - 12:00 AM Monday - Friday |
| | 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays |

If a dynamic option event occurs during a usage transaction period, the standard TOU map is overridden with the "dynamic" TOU map. In our example, a usage transaction period that includes August would use the dynamic TOU map when generating TOU map data for the time between 10:00 AM and 2:00 PM on August 2, but would use its standard TOU map for the rest of the month.

# About Dynamic Option Events

Dynamic Option Events are specific periods of time during which a dynamic option is applicable.

There are circumstances and conditions during which the rules for creating TOU map data might need to be calculated differently than according to the utility's standard rules. Examples of this might include critical peak periods, curtailment requests or other demand response events. During these types of events, the TOU rules defined for a TOU map may be overridden. This is done through the use of dynamic options and dynamic option events.

A dynamic option may have many dynamic option events over time.

# Maintaining Dynamic Options

This portal is used to display and maintain a Dynamic Option.

Refer to About Dynamic Options for additional functional information.

You can access the portal from the **Menu > Usage > Dynamic Option**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Dynamic Option**: defines the basic information about a Dynamic Option.
- **Dynamic Option Event List**: provides a list of the Dynamic Option Events that have been created for this Dynamic Option.

# Maintaining Dynamic Option Events

This portal is used to display and maintain a Dynamic Option Event.


Copyright © 2026, Oracle and/or its affiliates.

Refer to About Dynamic Option Events for additional functional information.

You can access the portal from **Menu > Usage > Dynamic Option Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zone may appear as part of the portal's **Main** tab page:

- **Dynamic Option Event**: defines the basic information about a Dynamic Option Event.

# Consumption Extract Requests

- About Consumption Extract Requests
- Maintaining Consumption Extract Requests

## About Consumption Extract Requests

Consumption Extract Requests are created every time measurement or interval data is extracted for Oracle Utilities DataConnect or via Configurable Consumption Extracts. The consumption extract request can be created manually, typically to initiate an initial load type of extract, or created automatically, typically for an incremental type of extract, based on a configuration on the consumption extract type.

The Consumption Extract Request is responsible for submitting and monitoring the actual batch jobs that perform the data extract. In addition, it can be used to store any additional filters to further trim down data that will be extracted.

Refer to the Oracle Utilities DataConnect section in the *Administrative User Guide* for more details on how consumption extract requests are used with DataConnect.

Refer to Configurable Consumption Extracts in the *Administrative User Guide* for more information about how they are used with configurable consumption extracts.

## Maintaining Consumption Extract Requests

This portal is used to display and maintain a Consumption Extract Requests.

Refer to About Consumption Extract Requests for additional functional information.

You can access the portal from **Menu > Usage > Consumption Extract Requests**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Request**: displays the main attributes of a Consumption Extract Request.
- **Request Log**: displays all log information related to the Consumption Extract Request.

# Contacts

- About Contacts
- Maintaining Contacts


Copyright © 2026, Oracle and/or its affiliates.

## About Contacts

Contacts are individuals or business entities with which a company has contact. Service Points can have associated contacts which define the individual or business entity that uses the service (electricity, gas, water, etc.) delivered at the Service Point. A contact is optional for Service Points, but Usage Subscriptions must reference contacts.

**Note**: The base-package name search on the 360 Degree Search looks for Usage Subscription-related contacts. Use the Service Point Query portal to find a Service Point using a Service Point-related contact.

## Maintaining Contacts

This portal is used to display and maintain a Contact.

Refer to About Contacts for additional functional information.

You can access the portal from **Menu > Usage > Contact**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Contact**: defines the basic attributes of the contact
- **Service Points**: This zone lists service points associated with the contact.

# Communications


- Device Events
- Activities
- Communications
- Completion Events
- Notification Suppressions


# Device Events


- About Device Events
- Processing Device Events


## About Device Events

Device events are developments that have taken place relative to a device, and can include power outages, power restorations, tampering alerts, command completions, among others.

Attributes used to define device events include the following:



Copyright © 2026, Oracle and/or its affiliates.

- **Device Event Date/Time**: the date and time of the event. This date/time is flexible in its interpretation. For single events it is the moment the event took place. For events that are paired, like a power up and a power down, it will represent the start of the event when received with the first of the pair and the end of the event when received with the second of the pair. For an event that includes the duration, where the device event end date/time is also supplied, this date/time represents the start date/time.
- **Device Event End Date/Time**: the end date and time of events with durations. This will only be populated if the device event date/time is also populated.
- In addition, device events also reference details specific to the head-end system that sent the event, including the following:
- **Sender**: the head-end system (defined as a service provider in Oracle Utilities Smart Grid Gateway) from which the event was sent.
- **External Sender ID**: the external ID for the head-end system that sent the event.
- **External Event Name**: the external, head-end-specific name for the event. This name is translated into a "standard" event name within Oracle Utilities Smart Grid Gateway.
- **External Source Identifier**: an identifier for the source of the event.

### Standard Event Names

When the system is loading a device event, it first determines which head-end system sent it, and then determines which device produced it. The system then maps the device event name sent by the head-end system to a standard event name that corresponds to the event. Standard event names are beneficial because the event names sent by head-end systems vary, even when the events are essentially the same. What one head-end system calls "Primary Power Down," another system may call "Last Gasp." Creating a standard event name, such as "Power Outage - Device," makes it easier to send the device events to an external system such as Oracle Utilities Network Management System. In this way, the external systems that receive the events do not need to be configured to accept all of the possible names for a single type of event.

Refer to Understanding Device Event Types for additional information about how the device events are mapped using standard event names.

## Processing Device Events

Device Events received from external systems are processed through two major stages:

- **Device Event Seeder**: This is the common interface point that receives all device event information and performs critical validations and conversions. During this process the device event seeder identifies the appropriate specific device event business object for further processing. The choice is made based on the originating external system and the type of event being received. For example, a device event for a "Last Gasp" would result in the paired event first device event being selected.
  If for some reason the device event seeder encounters an error an instance of the device event eeder will be created to manage the resolution of that error.
- **Specific Device Event**: Each specific device event has the appropriate logic for the type of event that has been received. A tamper event might create a service issue monitor while a "Last Gasp" might create an outage activity.

### Device Event Seeder Processing

The device event seeder receives device event data in a common structure, however, the content of the data (identifiers, date/times, etc) can vary depending on the system and the


Copyright © 2026, Oracle and/or its affiliates.

particular device that generated it. To accurately translate the external content to values understood by the system the device event seeder leverages the configuration of the service provider (aka head end or external system) and the device.

The service provider specifies the following inputs into device event seeder processing:

- **Identifier Types:** Each device event must identify the device the event has been generated from. The system supports a multitude of identifiers for the device. The service provider configuration specifies which identifier type is used for the device which enables the device event seeder to search for the correct destination for the device event. For example, a service provider might indicate that devices are identified by a serial number.
- **Date/Time Format:** The service provider configuration will identify whether the date/times that were sent included specific time zone designations. This allows the system to accurately handle any time zone conversions that may be necessary.
- **Processing Methods:** A service provider has the following processing methods that are used in the evaluation of the device event:
- **Device Event Mapping :** Maps a given external device event name to a standard event name. That standard event name is eventually used to identify the appropriate device event for specific processing.
- **Time Zone Translation:** Identifies the extendable lookup to be used in mapping the service provider specific time zone to the interval time zone.

The device specifies the following inputs into IMD Seeder processing:

- **Incoming Data Shift:** This value is either specified directly on the device or can be defaulted via fallback on the device type. It identifies whether the incoming date/time information has is "Shifted" meaning it has been adjusted for Daylight Savings Time (DST) when in effect, or it is "Not Shifted" which means it will never be adjusted for DST. This plays an important role in the date/time processing since all incoming device event date/times will be stored without an adjustment for DST.

With these inputs the device event seeder is able to translate the external identifiers into the appropriate identifiers to tie the device event to the originating device in the system..

### Device Event Seeder Critical Validations

For device events to continue on to specific device event processing they must pass the critical validations:

- The device must be identified.
- The device event must have an event date/time.

If either of the critical validations fails the device event processing will be halted and an instance of the device event seeder will be created

Note: There is special logic that exists for handling the situation where either the device event date/time or the device event end date/time falls on the duplicate hour that occurs when a time zone transitions out of daylight savings time.

### Device Event Seeder Time Zone Translation and Conversion

To provide the most accurate tracking of device events all device event data is stored in a single time zone without an adjustment for Daylight Savings Time (DST). As utilities can have multiple head end systems that each send date/time information in different ways and some utilities operate in multiple time zones the device event seeder contains functionality to convert incoming date/times from a source time zone into the time zone of Oracle Utilities Meter Data Management (as indicated by the base time zone)


Copyright © 2026, Oracle and/or its affiliates.

There are three components to identifying the source time zone of the incoming device event:

- **Service Provider IMD Import Date/Time Format:** When the device event date/times are sent in a format that identifies the time zone for each date/time the date/times will be delivered to the device event seeder already converted.
- **Incoming Data Shift:** This indicates whether the date/times will be adjusted for DST while it is in effect. A value of "Always in Local Time" indicates that the date/times are adjusted for DST. A value of "Always in Standard Time" indicates that the date/times are not adjusted for DST
- **Time Zone**: The time zone of the incoming data is identified using the follow order of precedence:
    1. The time zone as indicated by the device event data
    2. The service point time zone
    3. The device configuration time zone
    4. The base time zone

Once the time zone and the data shift of the incoming date/times is identified all date/times will be converted from that time zone and shift into the base time zone in standard time.

### Specific Device Event Processing

### Types of Device Events

#### Standard

Momentary device events, such as a tamper alert, are processed using the standard device event. These device events do not provide any special processing.

#### Paired Device Events

Device events with a duration are defined using paired event device events: paired event first and paired event last. With the first of the pair representing the start of the event, and the last of the pair representing the end of the event. Events of this type can be configured to create or complete activities that represent the event. For example, a "Last Gasp" event might create an outage activity that is completed when "Power Restoration" event is received.

When pairs of events arrive in rapid succession (such as a last gasp followed quickly by a power restoration), these "paired event" business objects are designed to prevent them from being sent to subscribing applications.

#### Communication Response

The communication response is a special type of event used for head end systems that respond to smart meter commands with device events. When received these events create a communication in that will initiate the completion of the originating smart meter command.

Refer to About Communication In for more information on a communication in.

#### Sending Device Events to Subscribing Systems

When device events are received, they are typically passed onto to another subscribing system, such as Oracle Utilities Meter Data Management, a customer information system (such as Oracle Utilities Customer Care and Billing), an outage system (such as Oracle Utilities Network Management System), or some other application.


Copyright © 2026, Oracle and/or its affiliates.

The means of sending device event information to subscribing system is defined in the "How to Send Device Event Related Information" processing method for the service provider representing the subscribing system. Device event information can be sent via outbound communication business object, outbound communication, or batch process.

The method for sending device events to subscribing systems (business object, outbound message, or batch process) can be defined for each device event category, and can be overridden for individual device event types (including the ability to exclude specific device event types within a category. In addition, a default event processing method can be also be configured that applies when methods of transmission aren't specified at the individual category level.

The "Subscribe to Device Event" business service is used to process device event subscription requests, and allow external applications to manage the categories of events they receive.

Refer to About Device Event Notification Suppressions for more information on a suppression device event notifications.

**Creating Service Issue Monitors and Service Investigative Orders**

Device events can be configured to create service issue monitors and service investigative orders. The service issue monitor type to create is defined on the device event type. When a device event is received, an Enter algorithm on the device event's Processed state creates the service issue monitor, based on the Service Issue Monitor Type specified for the device event type.

Refer to Understanding Service Task Types for more information on configuring service issue monitor types.

**Device Event Prioritization**

By default, device events are created and/or received into the system in an initial state is "Pending" and are transitioned to subsequent states via the batch processing. This batch processing can be skipped by setting the Execution Method flag on the device event to "Real Time" (D1RT), in which case an Enter algorithm transitions the device event to the next state in its lifecycle immediately. This feature can be used to prioritize processing of certain types of device events.

The Execution Method flag can be set in in unsolicited device events sent from the head-end system, or can be inserted during Oracle Utilities Smart Grid Gateway Adapter processing.

Oracle Utilities Smart Grid Gateway Adapters set this flag to "Real Time" on all device events created as the result of smart meter commands.

# Activities


- About Activities
- Maintaining Activities


## About Activities

Activities cover a range of functional process within the system from smart meter commands, synchronization of consumption between related measuring components, service order management activities, and statistics and aggregations.



Copyright © 2026, Oracle and/or its affiliates.

The table below provides the full list of activity type categories supported and a brief description of how they are used:

| Activity Type Category | Activities of this type category are used to |
|---|---|
| Bulk Activity | Initiate a smart meter command for a group of meters using Oracle Utilities Smart Grid Gateway |
| Command Request | Initiate a smart meter command for an individual meter using Oracle Utilities Smart Grid Gateway |
| Consumption Sync | Trigger related measuring component estimation synchronization |
| Device Event Activity | Represent a device event. These are used when viewing usage and events with the Oracle Utilities Meter Data Management 360 Degree View. |
| Dimension Scanner | Trigger dimension scanning when aggregating usage, statistics, and other data. |
| Error Activity | Represent errors in activity processing. |
| Extract Request | Retrieve usage and device events from a head-end system using Oracle Utilities Smart Grid Gateway |
| Field Activity | Initiate field work via a message to an external field work system. Activities of this type are most often created by a request orchestration activity. |
| Meter Read Download Activity | Download a list of meters to be read. |
| Multi Device Command Request | Initiate a smart meter command for multiple meters (for head-end systems that support multi-device commands) using Oracle Utilities Smart Grid Gateway |
| Non-Dispatchable Activity | Represent steps or stages in a larger process in which the system is waiting for a resolution from another system. Examples include "Wait for Measurement" or "Wait for Scheduled Read." |
| Orchestration Maintenance | Update or cancel an active service order activity. |
| Payload Statistics | Calculate processing statistics for usage and device event import processing |
| Request Orchestration | Orchestrate the steps necessary to fulfill a service request, based on the current state of the service point related to the request. |
| Suppression | Suppress sending of device events to subscribing systems (such as Oracle Utilities Network Management System) |
| Usage Transaction Correction Processor | Trigger correction processing for usage transactions when corrected usage is received that would impact an existing usage transaction |

# Maintaining Activities


Copyright © 2026, Oracle and/or its affiliates.

You use the **Activity** portal to view and maintain activities.

Refer to About Activities for additional functional information.

You can access the portal by selecting **Main**, **Communication**, **Activity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When creating new activities using this portal, if the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal will be available.

This portal includes the following zones:

- **Activity**: defines the basic attributes of the activity
- **Field Activity Attachments**: lists attachments related to the activity (displayed only if the current activity is a field activity)
- **Activity Hierarchy**: a tree that displays the activity and related activities (including field activities and smart meter commands), inbound and outbound communications, and completion events
- **Activity Related Completion Event**: lists completion events (if any) related to the activity

#### Manually Completing Service Order Field Activities

Service order field activities in the "Communication in Progress" state can be manually completed or canceled by clicking **Complete** in the **Record Actions** section of the **Activity** zone. This opens the **Field Activity Manual Completion** page where the user can enter appropriate details to manually complete the field activity (such as completion information, service point information, device information, and so on). When submitted, this page creates an inbound communication and transitions the field activity to the next state as appropriate.

Note: Implementations can define custom service scripts for each field task type in the **Manual FA Completion Controls** section on the Field Task Type extendable lookup to customize the sections displayed on the **Field Activity Manual Completion** page.

# Communications

- About Communication Out
- About Communication In
- Maintaining Communications

## About Communication Out

These transactions are typically created from an activity and are tasked with sending a message to an external system. They follow one of two communication patterns:


Copyright © 2026, Oracle and/or its affiliates.

- Two-way: these messages are sent to the external system without an expectation of an immediate response, or a response is received but it is only an acknowledgement of a successful receipt of the message. The true response to the message is expected to be received in a process detached from the communication out. For the full lifecycle of the communication to be complete a communication in must be received.
- One-way: these messages are sent to the external system and responded to in full within a single transaction. No communication in will be received.

The format of the message sent to the external system is typically driven by a processing method for that external system and the transportation method is controlled by the external system's message options.

If an error is encountered sending the message to the external system the communication out can perform automated retry of the messages for a configurable rate of retry and number of retry attempts.

For one-way communications the communication out will be responsible for creating the completion event.

Once the message has been communicated and a response received the outbound communication will update the initiating activity with the result.

> **Note**
>
> Refer to the following topics for additional information on communications functionality:
>
> - Understanding Processing Methods for additional information on how processing methods are configured.
> - See About Completion Events for details on completion events.
> - See External Systems in the Oracle Utilities Application Framework *Administrative User Guide* for additioal information on configuring communication between your system and external systems.

## About Communication In

These transactions represent a message sent from an external system and are received in two situations:

- In response to a communication that had been sent to the external system. This is the most common usage of the communication in.
- An unsolicited message received from an external system such as a real time event. Note: this is not a common pattern.
- Upon receipt the communication in will link itself to the communication out that initiated it. The response message will then be validated to ensure that it is technically and functionally correct.
- For many smart meter commands and field activities executed by service order management the communication in is responsible for creating the appropriate completion events. For example, if the response was for the On Demand Read smart meter command a completion event will be created that will generate the appropriate initial measurement data.


Copyright © 2026, Oracle and/or its affiliates.

- Similar to the communication out a communication in that encounters an error can perform an automated retry with a configurable rate of retry and number of retry attempts.

## Maintaining Communications

The Communication portal is used to maintain communication in and communication out records.

Refer to About Communication Out and About Communication In for additional functional information.

You can access the portal from the **Main > Communication > Communication**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Communication**: defines the basic attributes of the Communication
- **Communication Hierarchy**: a tree that displays the communication and related activities (including field activities and smart meter commands), inbound and outbound communications, and completion events.

# Completion Events


- About Completion Events
- Maintaining Completion Events


## About Completion Events

Completion events are used to implement the outcome of an activity. They are created by either the communication out (for one-way communication) or the communication in (for two-way communication). Completion events are executed by the activity orchestrating the communication with the external system.

Some examples of the tasks a completion event will perform are:

- Commissioning, decommissioning, connecting, disconnecting a smart meter
- Installing or Removing a meter or item
- Creating initial measurement data
- Updating a device or service point

## Maintaining Completion Events

The Completion Event portal is used to maintain completion events.

Refer to About Completion Events additional functional information.

You can access the portal from the **Main > Communication > Completion Event**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:



Copyright © 2026, Oracle and/or its affiliates.

- **Completion Event**: defines the basic attributes of the Completion Event
- **Completion Event Related Objects**: displays data related to the completion event such as a device, service point, or initial measurement.

# Notification Suppressions


- About Device Event Notification Suppressions
- Maintaining Notification Suppressions


## About Device Event Notification Suppressions

When a device event is processed by the system it can be sent to one or many external applications interested in being notified about the event. A prime example of this is the Outage Management System subscribing to any device events related to a meter either losing power or regaining power.

There are situations, such as a field worker performing maintenance on a meter or on a node that impacts many meters, that will result in a large amount of device events being created, processed, and distributed to subscribers.

Device event notification suppressions provides a way for a subscribing system, in anticipation of scheduled maintenance, to suppress notification of certain device events (power down, power up, tamper, etc) from one or more meters.

## Maintaining Notification Suppressions

The **Activity** portal is used to maintain notification suppressions records.

Refer to About Device Event Notification Suppressions for additional functional information.

You can access the portal from the **Main > Communication > Notification Suppression**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Activity**: defines the basic attributes of a notification suppression activity

# Settlement


- Understanding Market Settlement
- Viewing and Managing Settlement Processes
- Settlement Calculations
- Settlement Item Market Participants
- Market Allocations


# Understanding Market Settlement



Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

- About Market Settlement Processing

# About Market Settlement Processing

Market settlement is the process of deriving billing determinants to be used in supplier billing based on the results of a data aggregation processes.

The load settlement process also involves calculating imbalances. An imbalance is the difference between two load shapes representing different estimates of the same period, typically a scheduled load shape and an actual load shape. Imbalances are typically used when money has changed hands based on an estimate that has been improved or finalized by the settlement process. Because the second estimate has more accurate data, the allocations performed by the first process have to be modified.

Imbalance shapes can be further divided by applying bands and TOU components. Bands are error tolerances around zero that allow for differential pricing of the imbalance curve by how well or poorly the scheduled estimate matched the actual usage. Time of Use periods are often defined, allowing a higher penalty for faulty forecasting during higher-priced, on-peak periods. Whether an imbalance calculation is performed and how it is performed varies greatly from market to market.

The following sections provide details regarding some of the specific processing used in settlement processing.

- Importing Data
- Forecasting and Estimating Usage


Copyright © 2026, Oracle and/or its affiliates.

- Aggregating Data
- Unaccounted for Energy (UFE) Calculations
- Allocating Load via Market Products, Awards, and Contracts

## Importing Data

In addition to customer-level data such as meter readings or billed service quantities (which are used aggregation), settlement processing also involves other data that must be available to the application. For example:

- UFE calculations use both aggregated data and overall system, or zonal, load data
- Estimations and forecasting often use weather data
- Settlement billing requires pricing data

Data of these types can be imported into the system from external sources via a number of ways:

- For external sources that send data on a regular basis, data can be imported in a manner similar to how measurement data is received head end systems. See Processing Initial Measurement Data in this guide and Data Import - Message Driven Bean Configuration in the *Oracle Utilities Meter Solution Administrative User Guide* for more information.
- For external systems that send data upon request, data can be requested via measuring component data requests. Requests of this type can be either scheduled or performed manually. See Requesting Measurement Data in this guide for more information.
- For external systems that make data available for use but don't support sending the data directly, data can be imported manually in a comma separated values (CSV) format. See Uploading Measurement Data in this guide for more information.

## Forecasting and Estimating Usage

Some settlement processes, such as initial settlement, often have to be performed before actual usage has been received or calculated for service points participating in the settlement market. In these cases, usage must be estimated, and in some cases, forecasted before aggregation and other settlement calculations can happen. A forecast is the term used when estimating usage data for future dates based on historical or profile data. Backcasting is the term used for estimating data for dates in the past. These can also be based on historical and/or profile data.

Two specific methods used for forecasting and backcasting include:

- **Profile Estimation**: This method uses profile data that represents typical usage for the service point to estimate usage. In some cases, profile data can also be adjusted based on scalar usage for the same service point.
- **Proxy Day Estimation**: This process compares data between measuring components to identify an historical day that most closely matches the date for which data is needed and uses data from that historical day to estimate usage

## Aggregating Data

Aggregation involves summing real or assigned (profiled or deemed) interval data to a user-defined aggregation level for further processing. Aggregation may occur at a variety of levels for a variety of reasons, including reporting requirements and applying losses and other factors


Copyright © 2026, Oracle and/or its affiliates.

at an aggregated level up to the market participation level. Aggregation is typically used to build ground-up estimates of load for energy market participants for forecasting, scheduling, and settlement.

Aggregation is performed using the Dynamic Aggregation feature of the Oracle Utilities Meter Solution.

### Aggregation Levels

Because aggregation involves summing up interval data to a specified level for processing, it's important to understand what is meant by the term "aggregation level" Put simply, an aggregation level is any point along the distribution chain to which entities below that point are summed. For instance, if you were going to sum up all the accounts that are related to a specific bus, that bus would be an aggregation level. Similarly, if you were going to aggregate all the energy used by all the accounts for a given Energy Service Provider (ESP), that ESP would be an aggregation level. In general, aggregation is usually performed at a level below you on the distribution chain from generators to end users.

The entities and/or market participants that serve as aggregation levels are specific to the particular area and market in which your utility is operating, but can include the following:

#### Energy Service Providers (ESP)

Energy Service Providers (ESPs) sell power to customers. They may generate it and/or buy it from other sources. ESPs are also referred to as Retail Energy Providers (REPs) or simply suppliers.

#### Distribution Companies (DisCo)

A distribution company is responsible for the delivery of power to all customers in its service territory (Jurisdiction), regardless of who the customers have selected as their suppliers. The DisCo owns and operates the distribution lines that bring electricity into homes and businesses, and is responsible for metering, billing, and service calls. However, a DisCo may or may not be affiliated with a customer's power supplier. Distribution companies are also referred to as Operating Companies, Utility Distribution Companies (UDCs), or Local Distribution Companies (LDCs).

#### Schedule/Service Coordinators (SC)

Schedule Coordinators (or Service Coordinators) are responsible for creating schedules of energy usage that represent the estimated usage for a future time period (most often one day) for a given segment of the market. Schedules are most often based on results generated during the load profiling process, but can also be based on historical usage information and other data.

#### Independent System Operators (ISO)

An Independent System Operator (ISO) is a non-profit corporation that manages the regional power grid to coordinate the generation and transmission of energy. Its goal is to guarantee a safe and reliable power supply, and to ensure that all suppliers get nondiscriminatory access to the transmission system (while the ISO controls the power grid, commercial entities still own it). In the U.S.market, ISOs are regulated by the Federal Energy Regulatory Commission (FERC).

The various aggregation levels used in settlement processing are defined by Aggregation Groups, Measuring Component Sets, and Aggregation Measuring Components.

### Aggregation Dimensions


Copyright © 2026, Oracle and/or its affiliates.

In addition to aggregating data by aggregation levels, it is useful to be able to aggregate by other factors, such as customer class, voltage class, or loss class. These and other factors are referred as dimensions and are referenced on measuring component sets. Some common aggregation dimensions are described below. Note that these are provided as examples of common aggregation dimensions only.

Specific markets may require aggregation dimensions other than those listed here.

**Customer Class**

A customer class represents a ‘type’ of customer or account. Three common customer classes include residential, commercial, and industrial, but others are also prevalent. In addition, there may be multiple types of residential, commercial, or industrial classes, representing distinct types of accounts within each of those broad categories. For instance, residential customers might be divided into apartments, condominiums, and houses.

**Voltage Class**

A voltage class represents a group of customers or accounts based on the type of voltage used by the end users in that group. Voltage classes differ by market and region, but some common voltage classes include Transmission, Primary, and Secondary. Aggregating by voltage class is often performed for reporting purposes, to track usage of a given voltage type within the system.

**Loss Class**

A loss class represents a group of customers or accounts based on the type of loss (transmission loss, distribution loss, etc.) associated with those accounts. Loss is often related to voltage levels, and so correlations between loss classes and voltage classes are common. Aggregation by loss class is used when calculating losses in the overall settlement process, and allows different loss calculation algorithms to be used with different loss classes, as appropriate.

**Rate Class**

A rate class represents a group of customers or accounts based on pricing. Rate classes can be based on a number of characteristics, including usage, territory, or even voltage type.

These and other dimensions are typically defined as attributes or characteristics of usage subscriptions, service points, devices, or measuring components.

- Applying Losses

## Applying Losses

Markets typically operate using generation-level determinants, meaning that the individual metered entity data must be adjusted for system losses in order to compare the market participant load to the system load, market participant schedules, etc. Loss calculations can vary from simple to complex, ranging from an annual percentage by rate class or voltage level, to seasonal or time of use loss factors, to hourly dynamic loss factor calculations based on system conditions. Loss factors can be applied at the metered level or any aggregation level, as needed. In addition, losses can be applied to generators in markets where generation losses are calculated and used.

Applying losses to aggregated data is performed via value derivation algorithms. These are algorithms which apply a configured formula to aggregated measurement values. For instance, to apply a loss factor, an algorithm can retrieve effective-dated factor values for a loss factor and multiply each aggregated measurement value by the appropriate factor value. Value derivation algorithms can retrieve values from all types of factors, including multi-variable


Copyright © 2026, Oracle and/or its affiliates.

factors defined for settlement units. See Aggregation Calculation — Other Calculations for more information about how losses can be applied during data aggregation.

## Unaccounted for Energy (UFE) Calculations

Unaccounted for Energy (UFE) is calculated by comparing the known system load to the sum of all market participant estimates from the estimation, aggregation, and loss adjustment processes. Because the market participant estimates will never perfectly match the system load, these estimates are fine-tuned by allocating a portion of the total UFE to each estimate. Allocation methods differ, but in general each market participant is allocated a portion of the system UFE for each hour based on the ratio of that particular estimate's load in that hour to the total of all estimated loads in that hour. This process forces the total of all estimates to match the system load. Some markets use different UFE allocation methods, including differential allocation of UFE to interval metered loads apart from profiled loads, and adjusting the profile methodologies to minimize UFE where possible.

UFE calculations are performed as part of the settlement calculation process, and can be performed using the Measuring Component Set Calculation Rule settlement calculation rule defined in a Bulk Settlement Calculation Group executed via the Measuring Component Set Calculation (D1-MCSCA) batch process.

For smaller markets, UFE calculations can be performed using a chained sequence of Array Math and Vector and Service Quantity Math settlement calculation rules. At a basic level, this sequence looks like this:

1. Sum the market participant load from aggregation results (using Array Math)
2. Subtract the market participant load from the system load to derive the total UFE for the calculation period (using Vector and Service Quantity Math)
3. Allocate the total UFE to the various market participants as appropriate (using Array Math)

> **Note**
>
> The results of UFE calculations are also used in other settlement calculations. These three steps represent part of a larger set of calculations.

## Allocating Load via Market Products, Awards, and Contracts

Distribution companies operating in deregulated markets are often designated as the Provider of Last Resort (POLR) for customers that either have not selected a supplier or have been dropped by a supplier. If allowed by legal and market rules, the distribution company can elect to outsource the responsibility of supplying POLR load to qualified wholesalers.

The POLR consumers will still, either directly or indirectly, receive bills for distribution services from and pay the distribution company, but the wholesaler takes responsibility for procuring and serving the POLR load, for which they receive payments from the distribution company.

A simple example of the POLR allocation process might involve the following steps:

1. Subtract all other suppliers from the zonal load to get the remaining POLR load.
2. Allow for the distribution company to separate out any owned generation.
3. Divide POLR load amongst awarded wholesale contracts.

Details of these transactions would be sent to the market operator by contract ID.


Copyright © 2026, Oracle and/or its affiliates.

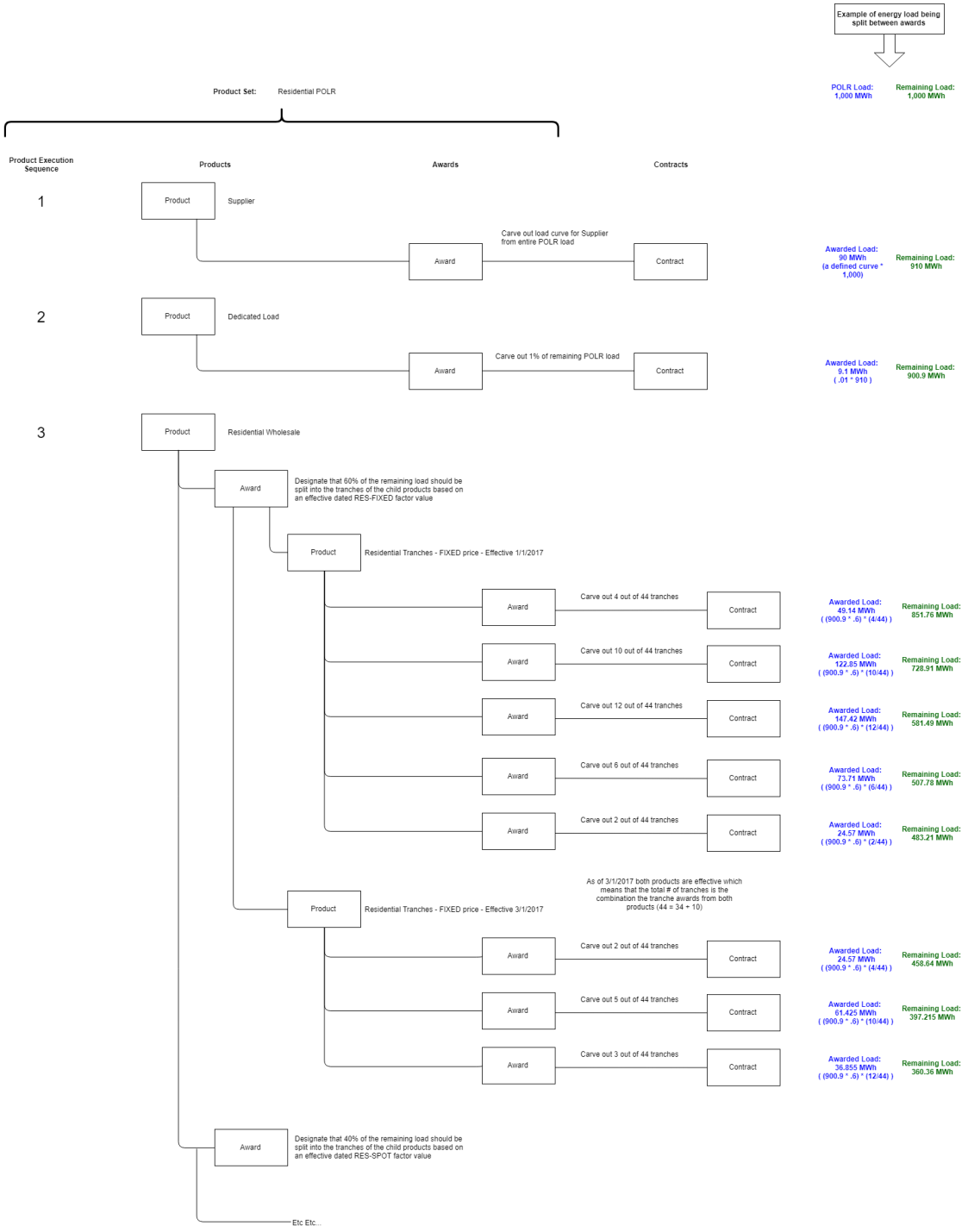
These awards of POLR load can be defined in the application by Market Products (which belong to a Market Product Set), Market Awards, and Market Contracts.

To elaborate on the example above, a market product set might the following set of market products:

- A product and award to carve out load from the POLR load for one or more suppliers. This would be a “Fixed Profile” award.
- A product and award to carve out a percentage of the remaining POLR load for owned generation. This would be a “Percentage” award.
- A product and awards that distribute the remaining POLR into various bundles of tranches (chunks of load) to wholesalers. These would be “Tranche” awards.

The following diagram illustrates the way in which this distribution would work.


Copyright © 2026, Oracle and/or its affiliates.




The Market Award Allocation settlement calculation rule is used to perform the allocation of load the designated market products and awards. This rule is part of a larger settlement calculation process, using settlement quantities (the POLR load) calculated by other rules (such as Array Math and Vector and Service Quantity Math). The Market Award Allocation rule allocates those quantities to the products within a specified Market Product Set, and saves the results for export to the appropriate market participants.

# Viewing and Managing Settlement Processes


Copyright © 2026, Oracle and/or its affiliates.

You can use the Process Flows feature of the Oracle Utilities Application Framework to create, view, and manage settlement processes. See Process Flows in the *Oracle Utilities Application Framework Administrative User Guide* for more information.

# Settlement Calculations


- About Settlement Calculations
- Settlement Subscriptions
- Settlement Transactions
- Settlement Items
- Settlement Item Quantities
- Settlement Billed Usage


## About Settlement Calculations

Oracle Utilities Market Settlements Management can calculate and publish settlement results calculated from aggregated data to market participants on an ongoing basis. Settlement calculations derive a settlement transaction's service quantities using the aggregated measurement associated with one or more market participants.

Before settlement results can be calculated, you must first create a Settlement Subscription. The settlement subscription defines the external system to which the data should be sent as well as the settlement calculation groups that will be used for calculation. The settlement calculation rules defined for the settlement calculation groups are responsible for performing the specific calculations, validations, or estimations for settlement results.

Settlement service quantities are derived from aggregated measurement data the calculation period.

The results of all calculations performed by settlement subscriptions are stored as a Settlement Transaction.

The settlement calculation process has a number of different objects that are detailed below that allow for configurability and flexibility in how the process executes.

**Settlement Calculation Groups**

Settlement calculation groups are collections of settlement calculation rules that are used to calculate service quantities for market participants. During the settlement transaction process, the system executes the settlement calculation rules defined in the settlement calculation group referenced on the settlement subscription. The rules within a settlement calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

For information on configuration, reference the related section for Configuring Settlement Calculation Groups in the Administrative User Guide.

**Settlement Calculation Rules**

The specific calculation processing performed on a settlement transaction is defined in individual settlement calculation rules, each performing a specific set of targeted logic. The base product contains many settlement calculation rules you can use in your implementation, but you can also create your own custom settlement calculation rules. Many settlement



Copyright © 2026, Oracle and/or its affiliates.

calculation rules are identical to usage calculation rules used by Oracle Utilities Meter Data Management

For a list of validation settlement calculation rules, reference the About Usage Validations section. For a list of estimation usage calculation rules, reference the About Usage Estimations section. For a list of all other usage calculation rules that handle calculation of bill determinants, reference the Inventory of Other Usage Calculation Rules section.

For information on configuration, reference the related section for Configuring Settlement Calculation Rules in the Administrative User Guide.

### Settlement Transaction Exceptions

Each settlement calculation rule defines an exception type and severity that specify how exceptions are tracked by the system. When a settlement transaction fails some part of a settlement calculation rule, an exception of the type specified for the failed settlement calculation rule is created. A single settlement transaction can have multiple exceptions, one (or more) for each rule that failed. This allows users to see all of the problems detected during the settlement calculation process.

There are three levels of severity for settlement transaction exceptions:

- **Information**: Used to highlight minor issues, but not sufficient to cause the settlement transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the settlement transaction from being exported. Multiple "issue exceptions" can be created during settlement transaction processing. If at least one issue exists after all rules have been applied, the settlement transaction is transitioned to a failure state requiring review and approval.
- **Terminate**: Used to report a severe issue that will cause the settlement calculation process to stop and the settlement transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop on for a settlement transaction). This should be used for cases where manual override / approval isn't accurate. For example, a "Curve Not Continuous" error that says the interval data doesn't cover the full settlement period should always be set to Terminate as an Exception Severity.

Note that exceptions are not deleted when a settlement transaction is adjusted or corrected. After any issues are corrected or the settlement transaction is overridden (or manually completed), the exceptions persist in a closed state for reporting purposes.

In addition to exceptions, settlement processing can also trigger the creation of To Do Entries related to issues. If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the settlement transaction is transitioned to the Exception state. The To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the settlement transaction.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

For information on configuration, reference the related section for Configuring Usage Transaction Exception Types in the Administration Guide.

### Settlement Calculation Rule Eligibility Criteria

Settlement calculation rule eligibility criteria are user-definable conditions that could cause a given settlement calculation rule to be applied or skipped. This can involve the evaluation of some attribute of the settlement subscription or something else entirely. A settlement calculation rule can have multiple eligibility criteria for determining if the rule should be applied


Copyright © 2026, Oracle and/or its affiliates.

or skipped, based on a user-defined sequence. Eligibility criteria for settlement calculation rules works in the same manner as eligibility criteria for usage calculation rules used by Oracle Utilities Meter Data Management.

For information on configuration, reference the related section for Configuring Usage Calculation Rules in the Administrative User Guide.

**Settlement Calculations and Anchor Time**

Settlement calculation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing settlement transactions. When processing settlement calculations based on dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.

**Note**: When processing calculations for measuring components whose type specifies an **Anchor Time**, if the Start Time and End Time of the calculation period do NOT match the **Anchor Time** on the measuring component type, the Start and Stop Time are shifted to align with the closest time to the **Anchor Time** that aligns with the interval size of the measurement data being processed. The Start Time is shifted backwards (to an earlier time) and the End Time is shifted forwards (to a later time). In the case of hourly intervals with Start Times and End Times on the hour, no shift occurs. In the case of daily intervals, the shift is to the nearest Anchor Time. For example, in the Anchor Time is 07:00AM but the Start Time and End Time are both 12:00, the Start Time would be shifted to the previous 07:00AM, and the End Time would be shifted the next 07:00AM.

# Settlement Subscriptions


- About Settlement Subscriptions
- Maintaining Settlement Subscriptions


## About Settlement Subscriptions

Settlement subscriptions track the ongoing need to calculate a set of data, often aggregated load, for the purpose of settling a market and send the results to a recipient.

Settlement subscriptions must reference the following:

- **Recipient**: The identity of the subscribing system. In other words, you must set up a recipient for any system that subscribes to settlement calculations.
- **Settlement Calculation Groups**: The calculation groups used to perform settlement calculations for settlement transactions based on this settlement subscription. Settlement calculation groups can be configured in a specific sequence of execution. In addition, each group can be configured to require approval before the calculation execution proceeds to the next calculation group.

Settlement subscriptions can also reference the following;

- **Factor Overrides**: One or more factors can be used as override values when performing calculations
- **Measuring Components**: One or more measuring components directly linked to the settlement subscription. These are most often aggregation measuring components, but can also include profile measuring components.



Copyright © 2026, Oracle and/or its affiliates.

- **Measuring Component Sets**: One or more measuring component sets directly linked to the settlement subscription.

## Maintaining Settlement Subscriptions

You use the portal **Settlement Subscription** portal to view and maintain settlement subscriptions.

Refer to About Settlement Subscriptions for additional functional information.

You can access the portal from the **Main > Settlement > Settlement Subscription**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Subscription**: This zone displays basic information about the settlement subscription, including factor overrides, settlement calculation groups, measuring components and measuring component sets.
- **Settlement Transaction**: This zone displays a list of settlement transactions for the settlement subscription.

# Settlement Transactions


- About Settlement Transactions
- Settlement Transaction Communication
- Maintaining Settlement Transactions


## About Settlement Transactions

Settlement transactions are records of calculations for a settlement subscription. All of the service quantities calculated are stored as part of the settlement transaction.

### Settlement Calculation Group Determination

The first step in settlement calculation is to determine the settlement calculation group holding the rules set used in calculating or validating the calculations. There are two ways of configuring the settlement calculation group to use in the processing. It may be specified on the settlement subscription during its creation with an effective and expiration date. If there are no settlement calculation groups in effect for the settlement transaction, the **Fallback Settlement Calculation Group** specified in the settlement subscription type will be used.

### Calculation Period

Settlement transaction requests must specify the date range for the settlement transaction. This date range is referred to as the calculation period for the settlement transaction.

### Date Breaks

Date breaks are used to break up a settlement period into sub-periods based on the dates on which changes took place for market participants. One approach to this situation would be for the system to crate settlement transactions (where each has the desired date range). Another



Copyright © 2026, Oracle and/or its affiliates.

approach is for the system to create single settlement transaction with date breaks that define the date ranges before and after the change.

If a settlement transaction has date break(s), the settlement calculation engine segregates the usage into multiple calculation periods based on the date breaks. If there are no date breaks in the settlement transaction, a single calculation period is created for the entire calculation period.

### Service Quantities

Every settlement period created for a settlement transaction contains one or more service quantities. Service quantities are calculated by the settlement calculation group's rules specified for the settlement subscription from which the settlement transaction is created. For example, a settlement transaction might have quantities calculated by a single rule that applies a TOU map to aggregated kWh .

Each service quantity lists details about that quantity, including a UOM (or TOU or SQI as appropriate) and a quantity. In addition, each service quantity also references the source (measuring component) of the quantities (for audit purposes).

### Quality Assessment

The Get Interval Data, Get Scalar Details, and Get Time of Use Mapped Usage calculation rules include a "Perform Measurement Quality Assessment" option that indicates (yes or no) if the measurement conditions of the measurements used when calculating usage should be evaluated and included in the resulting settlement transaction and outbound message. When enabled (set to "Yes"), the calculation rule algorithm evaluates the condition codes of the measurements used in the calculation to determine if they fall within a "Regular" range of conditions (this range is defined by the "Bottom Regular Condition Range" and "Top Regular Condition Range" algorithm parameters). The result of this assessment is displayed in the "Data Quality Assessment" column in the Calcuatio0n Periods section which indicates if the service quantity is based on "regular" or "non- regular" data. (If the "Perform Measurement Quality Assessment" option is disabled, the "Data Quality Assessment" column displays "No Assessment Done). In addition, the "Source Measurement Quality List" column contains XML that contains a list of measurements aggregated by condition/quality.

### Validations

As settlement transactions are created and move through their various states, several validations are performed to ensure proper processing. These include:

- Validating the business object used to create the settlement transaction
- Validating the start and end dates for the settlement transaction
- Validating the settlement transaction’s service quantities
- Validating the existing settlement transactions for the same calculation period
- Validating whether or not the settlement transaction can be deleted

### Creating Settlement Transactions Using Batch Processing

The Create Scheduled Settlement Transaction (D1-CSST) batch process can be used to generate settlement transactions based on a scheduled timetable.

The schedule for creating settlement transactions is defined based a Processing Timetable Type defined on the settlement subscription type. The process uses the details in the Processing Timetable Type (and its **Execution Frequency Class**) to determine the settlement transaction start and end dates. **Execution Frequency Class** options include:


Copyright © 2026, Oracle and/or its affiliates.

- **Daily**: The process runs based on the **Scheduling Information** parameters defined in the Processing Timetable Type .
- **Defined Dates**: The process runs for the most recent period defined in the **Processing Dates** section of the Processing Timetable Type that falls before the Batch Business Date. (but after the **Start Date** on the settlement subscription)

The business object to use for creating settlement transaction(s) is based on what is configured on the settlement subscription type. If there is no processing timetable type specified on a settlement subscription type, it will be skipped.

By default the process creates settlement transactions for all settlement subscription types. The optional "Restrict by Settlement Subscription Type" (restrictBySettlementSubType) parameter can be used to restrict processing to a specific Settlement Subscription Type.

Refer to Processing Timetable Types for more information about processing timetable types.

## Settlement Transaction Communication

Upon completion of calculations (or approval of calculations if the approval is required), settlement transaction results can be exported and sent to market participants This is done via an outbound communication created by the Create Settlement Transaction Outbound Communication algorithm. This communication is responsible for deriving and gathering the appropriate data to send to the market participant and creating and sending an outbound message.

There are two many ways in which settlement transactions are sent:

- **Contract**: Sending by contract involves sending the results of a settlement transaction to a recipient (such as an ISO) using an identifier from the market contract.
- **Identifier**: Sending by identifier involves sending the results of a settlement transaction to a recipient using an identifier from the market participant.

### Communication Types

The type of outbound communication created is based on the **Outbound Communication Type** defined on the settlement transaction's settlement subscription type. Communication types define the following about settlement transaction outbound communications:

- **Communication Flow**: The direction of communications based on this type (outbound or inbound).
- **Suppress Zeros**: Designates whether or not data should be sent for a service quantity of 0.
- **Message Generation Method**: Specifics if messages be sent as a single message to the market or broken down to one message per Identifier/Contract.
- **External Communication Type**: Defines whether the communication type is "Settlement by Contract" or "Settlement by Identifier".
- **Export Data Identification**: Defines combinations of UOM/TOU/SQI that represent the final outcome of the settlement transaction to be passed to the market
- **Reconciliation**: Defines the appropriate identifier or contract to include in the data sent to the market participant:
  - **Contract Reconciliation**: defines the appropriate identifier for the market contract associated with the service quantity. Used when sending data by contract.


Copyright © 2026, Oracle and/or its affiliates.

  - **Identifier Reconciliation**: Defines the characteristic type for the market participant used to retrieve the appropriate identifier. Used when sending data by identifier.
- **Exception Handling**: Defines how exceptions are handled, including To Do Type and Role, Retry Frequency, and the maximum number of times the system will retry to process the communication

#### Retrieving Service Quantities

Once created, the outbound communication retrieves a list of service quantities for the settlement transaction and inserts the details into the Settlement Transaction Details list. The specifics of how this is done is based on whether the settlement transaction is sent by contract or by identifier.

#### Send Settlement Transaction By Contract

Sending data by contract uses outbound communications based on the Send Settlement Transaction by Contract business object. The Retrieve Settlement Transaction SQs Contract algorithm retrieves the list of service quantities that match the UOM/TOU/SQI combination defined on the communication type, and for each service quantity found that has a market contract it will retrieve the identifier for that contract and populate the appropriate details into the Settlement Transaction Details list. The algorithm processes the entire service quantity result set even if it cannot find an identifier for a given contract. This way a user can see all identifier resolution errors at once (instead of needing to fix them one by one to see the next error).

#### Send Settlement Transaction By Identifier

Sending data by identifier uses outbound communications based on the Send Settlement Transaction by Identifier business object. The Retrieve Settlement Transaction SQs for Identifier algorithm extracts the market participant from the attributes associated to the service quantity's measuring component. The measuring components that will be associated to the service quantity in this instance will be aggregation measuring components that hold a list of dimensions that incudes a market participant. The algorithm then retrieves the list of service quantities that match the UOM/TOU/SQI combination defined on the communication type and for each service quantity found that has a measuring component, it retrieves the measuring component's market participant and then retrieve the identifier for that market participant and populate the appropriate details into the Settlement Transaction Details list. The algorithm processes the entire service quantity result set even if it cannot find a market participant or an identifier for a given market participant. This way a user can see all identifier resolution errors at once (instead of needing to fix them one by one to see the next error).

- Sending Settlement Transactions to Market Participants

## Sending Settlement Transactions to Market Participants

After the outbound communication has retrieved the appropriate service quantities, the outbound message creates an outbound message to the market participant (defined as the **Recipient** on the settlement transaction's settlement subscription or settlement subscription type).

The Send Settlement Transaction Details algorithm creates one more outbound messages based on the outbound message type defined for the market participant's Settlement Transaction Notification processing method (see below for more details).

- If the **Message Generation Method** on the communication type is set to "One Comprehensive Message" the outbound communication creates a single outbound message that contains all the service quantities retrieved.


Copyright © 2026, Oracle and/or its affiliates.

- If the **Message Generation Method** on the communication type is set to “Multiple Messages Based on Data” the outbound communication creates a separate outbound message for each of the service quantities retrieved.

### Configuring Market Participants for Settlement Transaction Communication

To support sending settlement transactions to market participants, the following processing method must be defined for the market participants to which settlement transactions are sent:

| Processing Role | Processing Method |
|---|---|
| Settlement Transaction Notification | How to Send Settlement Subscription Related Information<br>Used to send a settlement transaction outbound message to a market participant.<br>**Default Outbound Message Type**: The default outbound message type to create for outbound message sent to market participants. The base package does not include an outbound message business object for this. This must be created based on the requirements of each implementation and market participant.<br>If appropriate, override outbound message types can be defined for specific settlement subscription types. |

In addition, the following objects must be configured to send data requests:

- External System
- Outbound Message Type
- Message Sender

`

## Maintaining Settlement Transactions

You use the portal **Settlement Transaction** portal to view and maintain a settlement transactions.

Refer to About Settlement Transactions for additional functional information.

You can access the portal from the **Main > Settlement > Settlement Transaction**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Transaction**: This zone displays basic information about the settlement transaction, including processing information and settlement calculation groups.
- **Outbound Communication**: This zone displays a list of outbound communications sent for the settlement transaction.
- **Calculated Data Overview**: This zone displays an overview of calculated data related to the settlement transaction.
- **Calculated Data Details**: This zone provides ability to query the detailed service quantity interval data for the settlement transaction.

A number of important actions are available for a settlement transaction depending on its status:


Copyright © 2026, Oracle and/or its affiliates.

| Action | Allowed Statuses | Details |
|---|---|---|
| Approve | Approval In Progress<br>Issue Detected (only when the exceptions are marked to require approval) | Allows a user to bypass any exception or failure message for the settlement transaction and attempt to send it to the requesting system. |
| Calculate | Approval In Progress<br>Pending | Allows a user to trigger settlement calculations. |
| Discard | Approval In Progress<br>Export Error<br>Issue Detected | Allows a user to discard the settlement transaction. |
| Export | Export Error | Allows a user to transition the settlement transaction to the Export status and send the settlement transaction to the recipients. |
| Export Manually | Export Error | Allows a user to manually export the settlement transaction send it to the recipients. |
| Override Exception | Issue Detected | Allows a user to override an exception and transition the settlement transaction to the Override Exception state. |
| Re-Calculate | Issue Detected | Reruns the entire process of calculation along with any validations (all settlement calculation rules) for the settlement transaction. |
| Supercede | Completed | Allows a user to indicate that the settlement transaction was corrected after being sent. Sets the status of the settlement transaction to Superceded. |

### Batch Processing

The Create Scheduled Settlement Transaction (D1-CSST) batch process can be used to generate settlement transactions based on a scheduled timetable. Refer to Creating Settlement Transactions Using Batch Processing for more information about this batch process.

# Settlement Items

- About Settlement Items
- Maintaining Settlement Items

## About Settlement Items


Copyright © 2026, Oracle and/or its affiliates.

Settlement items represent end customer accounts. These are market participants for which settlement calculations are performed, and generally correspond to the market participants which subscribe to settlement calculation results via settlement subscriptions.

Settlement items are defined by the following:

- **Settlement Item Type**: The Settlement Item Type for the settlement item, which defines the business object to use to create the settlement item, the business object to use to capture billed usage for a settlement item (via Settlement Account Billed Usage), and valid service point types.
- **Division**: The division to which the settlement item belongs
- **Access Group**: The access group to which the settlement item belongs (used to limit access to records)
- **Status**: The status of the settlement item
- **Start Date/End Date**: The date range during which the settlement item is active.
- **Main Contact**: The main contact for the settlement item.
- **Time Zone**: The time zone in which the settlement item is active.
- **Measuring Components**: One or more measuring components directly linked to the settlement item. These are most often aggregation measuring components, but can also include profile measuring components.
- **Rate Class**: One or more effective-dated rate classes for the settlement item.

## Maintaining Settlement Items

You use the portal **Settlement Item** portal to view and maintain a settlement items.

Refer to About Settlement Items for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Settlement Item**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item**: This zone displays basic information about the settlement item, including related service points and measuring components.
- **Settlement Item Quantities**: This zone displays a list of settlement item quantities. See About Settlement Item Quantities for more information. To add a new quantity, for the settlement item, click **Add** in the zone title bar.
- **Settlement Item Market Participants**: This zone displays a list of settlement item market participants for the settlement item, via a Settlement Item Market Participant. To add a new market participant for the settlement item, click **Add** in the zone title bar.
- **Characteristics**: This zone displays a list of characteristics related to the settlement item.
- **Identifiers**: This zone displays a list of identifiers used with the settlement item.
- **Interval Measurements**: This zone displays a list of interval measurement data snapshots associated with the settlement item. To add a new interval measurement data snapshot for the settlement item, click **Add** in the zone title bar.
- **Scalar Measurements**: This zone displays a list of scalar measurement data snapshots associated with the settlement item. To add a new scalar measurement data snapshot for the settlement item, click **Add** in the zone title bar.


Copyright © 2026, Oracle and/or its affiliates.

# Settlement Item Quantities

- About Settlement Item Quantities
- Maintaining Settlement Item Quantities

## About Settlement Item Quantities

Settlement item quantities are quantities that are stored at the settlement item level. These quantities are calculated at irregular frequencies (such as annually or quarterly) and thus do not make sense as a measured quantity on a device and measuring component. For example, in some deregulated markets the Peak Load Contribution could be stored here and effective dated over time to represent the changing contribution of a given settlement item to the market wide coincidence peaks.

Settlement item quantities are defined by the following:

- **Settlement Item**: The settlement item for which the quantities were calculated
- **Settlement Item Quantity Type**: The Settlement Item Quantity Type used for the record
- **Start Date** / **End Date**: The date range for the calculated quantity
- **Settlement Item Quantities**: Calculated quantities based on **Settlement Item Quantity Identifiers** configured on the Settlement Item Quantity Type (see Understanding Settlement Item Quantity Types for more information)

> **Note**
>
> There cannot be more than one Settlement Item Quantity for the same Settlement Item and Settlement Item Quantity Type, for the same time period (defined by the Start Date and End Date parameters).

## Maintaining Settlement Item Quantities

You use the **Settlement Item Quantity** portal to view and maintain settlement item quantities.

Refer to About Settlement Item Quantities for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Settlement Item Quantity**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Quantity**: This zone displays the basic attributes of the settlement item quantity

# Settlement Billed Usage

- About Settlement Billed Usage


Copyright © 2026, Oracle and/or its affiliates.

- Maintaining Settlement Billed Usage

## About Settlement Billed Usage

Settlement billed usage represents billed usage for a settlement item. Settlement billed usage is calculated via settlement calculations performed for the settlement subscription associated with the settlement item. Settlement billed usage typically corresponds to settlement transactions.

Settlement billed usage records are defined by the following:

- **Settlement Item**: The settlement item for which the billed usage was calculated
- **Start Date/Time**: The date and time of the start of the billing period for the billed usage
- **End Date/Time**: The date and time of the end of the billing period for the billed usage
- **Billed Usage Periods**: one or more billed usage periods for the current settlement billed usage. A billed usage period is a period of time for which usage is calculated, including details calculated for each billed usage period such as UOM/TOU/SQI and quantities

## Maintaining Settlement Billed Usage

You use the portal **Settlement Billed Usage** portal to view and maintain settlement billed usage.

Refer to About Settlement Billed Usage for additional functional information.

You can access the portal from the **Main > Settlement > Settlement Billed Usage**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Billed Usage**: This zone displays basic information about the settlement billed usage, including billed usage periods.

# Settlement Item Market Participants

- About Settlement Item Market Participants
- Maintaining Settlement Item Market Participants

## About Settlement Item Market Participants

Settlement item market participants connects a Market, Market Participant and Settlement Item together over a specified date range. Settlement item market participants can be created either via the **Settlement Item Market Participant** portal, or by adding a market participant to a settlement item via the **Settlement Item** portal.

Settlement item market participants are defined by the following:

- **Settlement Item**: The settlement item for which this settlement item market participant was created.


Copyright © 2026, Oracle and/or its affiliates.

- **Market Relationship Type**: The market relationship (Distributor, Supplier, etc.) of the market participant and its role within a market.
- **Start Date/Time**: The start date/time for when the record is applicable.
- **End Date/Time**: The end date/time for when the record is applicable.
- **External ID**: An ID for the market participant used by external systems.
- **Market Participant**: The market participant associated with the **Settlement Item**, including an optional ID used by the market participant to uniquely identify the settlement item.

## Maintaining Settlement Item Market Participants

You use the **Settlement Item Market Participant** portal to display and maintain settlement item market participants.

Refer to About Settlement Item Market Participants for additional functional information.

You can access the portal by selecting **Main**, then **Settlement**, then **Settlement Item Market Participant**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Market Participant**: This zone displays the basic attributes of the settlement item market participant.
- **Sub Settlement Items**: This zone lists settlement items for the settlement item market participant.

# Market Allocations

- Market Contracts
- Market Awards
- Market Products

## Market Contracts

- About Market Contracts
- Maintaining Market Contracts

### About Market Contracts

Market contract are agreements with the market operator (such as an Independent System Operator, or ISO) related to how settlement calculation results are sent to the market. Market contracts are related to a specific market, have a seller and buyer, as well as a date range for when they are effective.

Market contracts are defined by the following:


Copyright © 2026, Oracle and/or its affiliates.

- **Market Contract Type**: The Market Contract Type for the contract. This defines the market in which the contract is in effect and the business object used to create the contract.
- **Contract** : A unique code for the contract, often provided by an external system
- **Start Date** / **End Date**: The date range during which the contract is in effect
- **Seller**: The market participant that serves as the Seller for the contract.
- **Buyer**: The market participant that serves as the Buyer for the contract.

## Maintaining Market Contracts

You use the **Market Contract** portal to view and maintain market contracts.

Refer to About Market Contracts for additional functional information.

You can access the portal from the **Main > Market > Market Contract**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Contract**: This zone displays basic information about the market contract.
- **Related Market Awards**: This zone displays a list of market awards related to the market contract.

# Market Awards

- About Market Awards
- Maintaining Market Awards
- Maintaining Market Allocations

## About Market Awards

Market awards are used to allocate load to various market participants.

Market awards are defined by the following:

- **Market Product**: The Market Product associated with the award
- **Award Class**: The type of award. The base package supports three award classes: Tranche Awards, Percentage Awards, and Fixed Profile Awards.
  - **Tranche Awards**: Tranches are used to handle a whole number division of load. For example, an agreement may be reached that the Provider of Last Resort (POLR) load for a zone should be broken into 64 tranches, or exactly 64 chunks of load that can be purchased. In this example, the load could be broken up in the following manner: Contract #1234 buys 12 tranches, Contract #1235 buys 1 tranche, and Contract #1236 buys 51 tranches (each of these three would be a market award) A factor can be used to define different tranche quantities over time for the same market award.
  - **Percentage Awards**: Percentage Awards allocate a percentage of the remaining load. A factor can be used to define a different percentage over time for the same market award.


Copyright © 2026, Oracle and/or its affiliates.

  - **Fixed Profile Awards**: Fixed Profile Awards leverage a profile measuring component to designate an amount that must be allocated from the overall load to this market award. A factor can be used to define different profiles over time for the same market award.
- **Start Date / End Date**: The date range for the market award
- **Award Details**: Details of the award, including the source (Static Value or Factor) and quantity (either a fixed value or factor value)
- **Characteristics**: Characteristics for the award, if appropriate

## Maintaining Market Awards

You use the **Market Award** portal to view and maintain market awards based on Market Products with a **Product Class** of “Contract Fulfillment”.

Refer to About Market Awards for additional functional information.

You can access the portal from the **Main > Market > Market Award**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Award**: This zone displays basic information about the market award, including **Award Details**, related market **Contracts**, and **Characteristics**.

## Maintaining Market Allocations

You use the **Market Allocation** portal to view and maintain market awards based on Market Products with a **Product Class** of “Parent Allocation”.

Refer to About Market Awards for additional functional information.

You can access the portal by selecting **Main**, then **Market**, then **Market Award**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the **Market Allocation** portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Allocation**: This zone displays basic information about the market award, including **Award Details**, and **Characteristics**.

# Market Products


- About Market Products
- Maintaining Market Products


## About Market Products

Market products are used to allocate market awards to market participants. Market products are referenced by the Market Award Allocation settlement calculation rule via a parent Market Product Set. Market products of class “Parent Allocation” can have child products. This allows definition of hierarchies of market products and related awards,



Copyright © 2026, Oracle and/or its affiliates.

Market products are defined by the following:

- **Market Product Set**: The Market Product Set to which the Market Product is associated.
- **Market Product Type**: The Market Product Type for the market product. This defines the business object used to create the market product and its market awards. For formula-based products, the market product type also defines the formula used to calculate award allocation.
- **Product Code**: A unique code for the product, often provided by an external system
- **Product Class**: The type of product. The base package supports two product classes: Parent Allocation or Contract Fulfillment:
  - **Parent Allocation**: Indicates that the market product is a parent level product that divides out a certain allocation of the total amount to be then divided further amongst child products.
  - **Contract Fulfillment**: Indicates that the market product ties to an "end result" award that can then be tied to a Market Contract
- **Execution Sequence**: Defines the sequence in which the market product will be executed within its parent Market Product Set
- **Awards**: One or more Market Awards associated with the market product. These define how load and procurement are allocated when the market product is processed by the Market Award Allocation settlement calculation rule.

## Maintaining Market Products

You use the **Market Product** portal to view and maintain market products.

Refer to About Market Products for additional functional information.

You can access the portal from the **Main > Market > Market Product**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Product**: This zone displays basic information about the market product.
- **Awards**: This zone displays a list of market awards related to the market product.

# Integrations

- Business Flags
- Analytics Integrations

# Business Flags

- About Business Flags
- Maintaining Business Flags

## About Business Flags


Copyright © 2026, Oracle and/or its affiliates.

Business flags provide a view into the status of a service point across applications (such as Oracle Utilities Customer Cloud Service, Oracle Utilities Meter Data Management, Oracle Utilities Customer Care and Billing and so on). Business flags are currently available in two forms:

- **Service Point Business Flags**: are often informational in nature but also allow for manual investigation to be taken.
- **Service Point Monitor Business Flags**: are seen as symptomatic of an issue that needs to be investigated. These business flags feed into a service issue monitor and may result in a service investigative order being created.

For additional details see About Business Flags in the Oracle Utilities Application Framework *Administrative User Guide*.

#### Business Flag Allowed Actions

Allowed actions are implemented presented to the user as selectable options that will guide the user through a particular method of investigating or resolving the business flag. The following actions are supported:

- **Create Field Activity**: This will initiate a field activity associated to the business flag's service point. The type of field task will be selected by the user. Note: this option requires Oracle Utilities Service Order Management.
- **Request Customer Notification:** This will initiate a message to the appropriate external system (e.g. Oracle Utilities Customer Care and Billing) that will identify a notification to be sent to the customer. The type of communication will be determined by the notification type selected by the user.

Note: not all business flags support allowed actions and not all business flags that support allowed actions support all allowed actions.

## Maintaining Business Flags

Information for this functionality is available in the Business Flags section in the Oracle Utilities Application Framework *Administrative User Guide*.

# Analytics Integrations


- Using the Meter Insights Investigation Viewer


## Using the Meter Insights Investigation Viewer

The Meter Insights and Investigations functionality leverages the Oracle Utilities Analytics Insights business flag integration to support receipt of revenue protection events (referred to as "insights") from Oracle Utilities Analytics Insights for research and investigation using Oracle Utilities Meter Data Management (or Oracle Utilities Meter Solution Cloud Service). Revenue protection insights sent from Oracle Utilities Analytics Insights for investigation are events that suggest potential theft or loss of revenue such as unexpected consumption from an inactive meter, evidence of meter tampering, discovery of an improperly installed meter (such as a meter installed upside down or in reverse), and so on. Insights received from Oracle Utilities Analytics Insights create business flags, which in turn create related Meter Insight Investigation service tasks (referred to as “investigations”). Business users can use the **Meter Insights Investigations Viewer** portal to search, view, and investigate these investigations. This portal



Copyright © 2026, Oracle and/or its affiliates.

is intended to provide as much investigative information as possible so as to allow the user to complete as much work as possible without leaving the portal.

### Searching

Use the following procedure to search for investigations:

1. Navigate to the portal by selecting **Main**, then **Insights**, then **Meter Insights Investigation Viewer**.

   By default the **Investigations** zone displays search results for unassigned records and records assigned to the current user.

2. To search by specific search criteria or display the search criteria and filters, click the **Show Filters** link or icon. The **Investigations** zone expands to display the available search filters, including:
   - Business Flag Type
   - Address (including City and Postal Code)
   - Average Daily Usage
   - Priority
   - Service Type
   - Service Point Type
   - Assigned to User ID
   - Flags to include Unassigned records, and investigations with current active field activities.
3. Enter your search criteria and click **Search**.

The **Investigations** zone displays a default set of data about each investigation. To select the information to display for each, click the **Display** link or icon, select or deselect the data you wish to view or hide, and click **Display**.

### Viewing Service Issue Monitors and Consumption

To view additional information about an investigation, click the **Expand Section** icon in the far left column. The section expands to display:

- Comments
- Service Point Information
- Field Work Information
- A list of insights (business flags) related to the investigation
- Consumption (a section similar to the **Final Values Overlay** zone in the **360 Degree View** portal)

To view additional information about the investigation, service point, or field activity, click the icon at the end of its information string. A new browser window will open displaying the **Service Task**, **Service Point** tab of the **360 Degree View** portal or **Activity** portal (respectively). This allows the user to investigate the issue in more detail as needed without having to navigate away from the **Meter Insights Investigations Viewer** portal.

To expand the **Consumption** section, click the **Expand Section - Consumption** icon on the far right side of the **Consumption** section heading. The section expands to display the following:


Copyright © 2026, Oracle and/or its affiliates.

- **Parameters**: The default parameters used to display consumption for the service point. This includes **Measuring Component**, **Value Identifier Type**, **Start Date/Time**, **End Date/Time**, **Final Values Overlay Profile**, and **Event Bar Profile**. These values are based on the **Consumption Data Source** parameters on the investigation's **Investigation Type** (Service Task Type). The measuring component and value identifier type is based on service point type, interval/scalar flag, and UOM/TOU/SQI combination, and the date range is based on the start and end dates of the investigation and the **Days Before to Graph** and **Days After to Graph** parameters.
- **Values to Graph**: A list of measuring components to graph, including measuring components defined in the current **Final Values Overlay Profile**, as well as other measuring components selected by the user.
- **Measurement Data**: A graph and tabular display of the consumption and values to graph, and event bars based on the current **Event Bar Profile**.

To change the displayed consumption, adjust the **Parameters** and **Values to Graph** and click **Load** (in the **Parameters** section).

#### Working With Investigations

Users can also add comments, assign a recipient for To Do entries, and change the status of investigations from this portal.

- To add or edit a comment, click the **Edit** icon in the second column from the left. The **Edit** icon is replaced by a checkmark icon and an "X" icon, and a **Comments** box opens. Enter or edit the comment text and click the checkmark icon. To delete a comment, click the "X" icon.

  Comments are saved on the Investigation, and can be viewed from the **Log** tab on the **Service Task** portal.
- To assign a recipient for To Do entries, click the **Edit** icon in the second column from the left. The **Edit** icon is replaced by a checkmark icon and an "X" icon, and the **Assigned To** column displays a drop-down list. Select the user to which To Do entries should be assigned, and click the checkmark icon.
- To change the status of the investigation, click the icon in the **Action** column. A popup menu will open displaying available options including:
  - Initiate Field Work: Select this to create a field activity to investigate the issue from the field (this transitions the investigation to the Waiting for Field Work state).
  - Issue Resolved: Select this if the issue has been resolved (this transitions the investigation to the Issue Resolved state).
  - No Issue: Select this if there is no issue (this transitions the investigation to the No Issue state).
  - Inconclusive: Select this if/when the investigation is inconclusive (this transitions the investigation to the Inconclusive state).
  - Discarded: Select this to discard issue (this transitions the investigation to the Discarded state).

See Meter Insights and Investigation Processingin the *Administrative User Guide* for more information about what happens when the investigation enters one of these states.

# Aggregations


Copyright © 2026, Oracle and/or its affiliates.

- Standard Aggregations
- Dynamic Aggregation

# Standard Aggregations

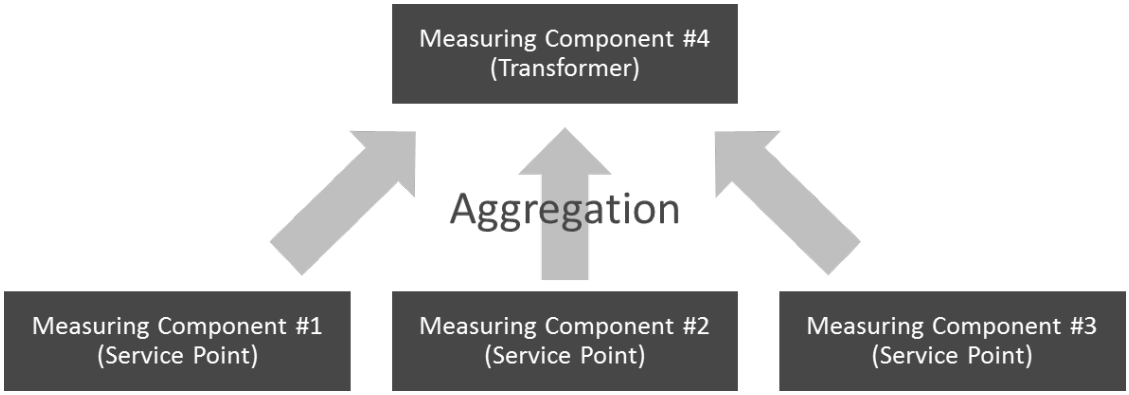
- About Aggregations
- Creating Aggregators
- Base Product Aggregators
- Maintaining Aggregations

## About Aggregations

Aggregations are measurements that represent a summarization of other measurements from a set of devices, items, and/or measuring components. For example, an aggregation may be based on the sum of the electric consumption of all residential customers in a particular postal code within the utility's service territory. The group of related measuring components for which measurements are summarized are referred to as constituent measuring components.

Every type of aggregation has one or more dimensions. For example, an aggregation might be defined with three dimensions as follows:

- Substation
- Feeder
- Transformer




Aggregations are defined by aggregator measuring components. A separate aggregator measuring component should be defined for every distinct combination of dimensions for a given type of aggregation. For example, the transformer aggregation described above would require an aggregator measuring component for every distinct combination of substation / feeder / transformer found among the electric Service Points.

Aggregations can also be aggregated. For example, the aggregator measuring components for each substation / feeder / transformer combination can in turn be used to aggregate consumption at each substation / feeder combination. In this case, a distinct aggregator measuring component would be defined for each unique substation / feeder combination. Likewise, the aggregator measuring components for each substation / feeder combination could be used to aggregate consumption at each substation, in which case, distinct aggregator measuring components would be defined for every unique substation. Finally, the aggregator


Copyright © 2026, Oracle and/or its affiliates.

measuring components for each substation can be used to aggregate total consumption for all substations, with an aggregator measuring component defined to represent the entire service area.

### Aggregation Calculations

The system periodically aggregates consumption via batch process, using a deferred monitor on the aggregator measuring components. In addition, users can re-aggregate data in real-time if they don't wish to wait for the batch process, or if the original aggregation needs to be re-calculated due to incorrect data. Users can also create ad hoc aggregations "on the fly" and these will persist in the database in the same manner as other aggregations.

### Understanding Aggregation Periods

The start and end dates and times for aggregation calculations are based on the following:

- Aggregation Horizon
- Aggregation Lag
- Aggregation Cut Off Time

Whenever aggregation is performed for an aggregator measuring component, consumption is aggregated for every day in its "Aggregation Horizon." The "Aggregation Horizon" is the number of days during which there's a potential change in measurement data for one or more of the measuring components associated an aggregator measuring component.

Aggregation calculations typically lag behind the current date by a few days to give the system time to upload and perform validations and create final measurements. The amount of lag time is referred to as the "Aggregation Lag" and is the number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the end of the aggregation horizon that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.





Aggregation is always performed through a given "through time" (such as 12:00 AM) rather than through the actual time of the aggregation calculation. This time is referred to as the "Aggregation Cut Off Time." For example, the stop time for aggregation calculations with an Aggregation Lag of 2, and an Aggregation Cut Off Time of 10:00 PM will always be 10:00 PM 2 days prior to the date on which the calculations are performed. In the above examples, aggregations performed on January 9 would have an end date/time of 10:00 PM on January 7, and aggregations performed on January 10 would have an end date/time of 10:00 PM on January 8.

### Aggregation and Re-Aggregation

The use of the aggregation horizon means that aggregated totals for some days will be re-aggregated until those days are no longer covered by the aggregation horizon. For example, with an aggregation horizon of 5 days and an aggregation lag of 2 days, on the night of January 9 the aggregation period would be January 3 through January 7 (as in the above example).


Copyright © 2026, Oracle and/or its affiliates.

On the night of January 10, the horizon will shift 1 day (to January 4 through January 8, again as above). This means the following:

- The totals for January 3 calculated on January 9 will be untouched (January 3 now falls outside the aggregation horizon)
- The totals for January 4 through January 7 will be re-derived because corrections may have occurred (and they still fall within the aggregation horizon)
- The totals for January 8 will be calculated for the first time (because it now is within the aggregation horizon).

### Manually Read Meters and Aggregation Lag

Aggregations that include manually read meters often have much longer aggregation lags than those for automatically read meters. This allows more time for manual meter readings to be imported into the system for use in aggregations.

For example, suppose a situation where manual meter reads arrive approximately 1 month after the date of the reading. In this case, it wouldn't be until a meter read upload on February 7 that the last manual reads including consumption from January 6 will exist. Since we don't want to perform aggregations for January 6 until there's a decent chance that all consumption for that date exists, an aggregation lag of 32 days ensures that the data for January 6 is in the system when the aggregation is performed on February 7.



### When An Aggregator Measuring Component Contains Both Manual and AMI Measuring Components

The previous section suggests that if an aggregator measuring component contains both manually read and AMI measuring components, the aggregator measuring component should have a long lag time rather than the short time shown in the earlier example.

While this is one possible approach, an alternative approach could allow users to see timely aggregations of the AMI channels and only lag the manual channels. To do this they could configure the system as follows:

- Create an aggregator measuring component that only aggregates manual measuring component types (horizon: 5 days, lag: 31 days)
- Create a separate aggregator measuring component that only aggregates interval measuring component type: 5 days types (horizon: 5 days, lag: 2 days)
- Create a third aggregator measuring component that aggregates the above aggregator measuring component types (horizon: 5 days, lag 31 days)

### Expanding The Aggregation Horizon Periodically

Even with a 30-ish day aggregation lag, it's possible for an account with several months of estimated consumption to have its consumption "invalidated" when real readings arrive whose dial readings are less than the estimated readings (or when major retroactive data change takes place). Another similar situation is when a user manually corrects consumption after the aggregation horizon has moved forward.

To address situations like these, the use of ad-hoc aggregations with a user-specified aggregation horizon supports the notion of periodically (such as once a month or once a week) aggregating with a long aggregation horizon (up to 90 days) to catch as many retroactive changes as possible.

Copyright © 2026, Oracle and/or its affiliates.

### A Note About Using Non-Effective Dated Dimensions

Be aware that whenever aggregation takes place, the system uses master data that exists as of that date. For example, if an aggregation is performed using non effective-dated master data attributes (for example, aggregating by measuring component type) and a measuring component's type is changed, all days in the aggregation horizon will reflect the change regardless of when the user makes the change, because the measuring component type relationship is not effective-dated.

The impact of this depends on the nature of the change:

- If the master data was wrong, this provides a way to re-aggregate the data to accurately reflect the master data.
- If the change occurred effective as of a given date (but there's no effective dated attribute), this can result in incorrect aggregation values.

Because of this, it is recommended that where possible dimensions used in aggregations be effective-dated.

For information on configuration, reference the related section for Aggregations in the Administration Guide.

## Creating Aggregators

There are two ways in which aggregator measuring components are created:

- Automatically: The system will periodically create new aggregator measuring components when new "dimensional values" are detected. For example, if a new substation is added to the system, a new aggregator measuring component will be created for the substation.
- Manually: An end-user can create an aggregator measuring component at their discretion

### Automatic Creation of Aggregator Measuring Components

The system automatically creates aggregator measuring components when it detects new instances of the dimension(s) being aggregated. This activity is referred to as dimension scanning. For example, if the system is configured to aggregate individual substations, a user will NOT have to manually set up an aggregator measuring component when a new substation if referenced on a service point. Rather, the system will create the new aggregator when it detects a substation on a service point that doesn't have an aggregator measuring component.

Similarly, if is configured to aggregate distinct combinations of rate class (on usage subscription type) and postal code (on service point), the system will create an aggregator measuring component when it detects new combinations of rate class and postal code.

The periodic monitoring to ensure sufficient aggregation measuring components exist is implemented via monitoring algorithms on a related activity business object that references a corresponding measuring component type. This means that there must be a separate activity type / activity combination for every type of aggregation performed. These monitor algorithms include:

- A Monitor algorithm on the Active state that simply transitions the activity to the transitory Scan state. Users can manually transition the activity to the Scan state by clicking the "Scan" button on the activity.
- An Enter algorithm on the Scan state finds the distinct combinations of dimensional values for this type of aggregation and checks that an aggregator measuring component that references the activity's aggregator measuring component type exists for every instance


Copyright © 2026, Oracle and/or its affiliates.

and creates new aggregator measuring components as appropriate. The algorithm also populates the dimensional values on the new measuring components (mapped to searchable characteristics), and sets the Consumption Aggregated Through Date/Time field on the new aggregator measuring components to the current date and the Aggregation Cut Off Time on the aggregator measuring component type.

### Hard-Wiring Service Types for Aggregations

If the dimensions used for aggregation do not include at least one service type-specific dimension (for example, in the case of aggregating by postal code only) AND the aggregation should not commingle measurements of different service types, it's important to:

- Declare the specific service type for the aggregation on the dimension scanner activity instance
- Declare the desired service type-specific measuring component types on the aggregator measuring component type

For example, if an electric and gas implementation aggregates consumption by postal code, they will need 2 activities: one will reference the Electric service type and the desired aggregator measuring component type, the other will reference the Gas service type and its aggregator measuring component type. The aggregator measuring components will have two dimensional attributes: postal code and service type.

### Manual Creation Of Aggregator Measuring Components

Users can also create aggregator measuring component manually. This is useful when an ad hoc aggregation for an ad hoc time period is required, or when there is no activity that automatically creates instances of a given combination of dimensions. To do this, the user simply clicks the "Add" link on the Aggregator Search zone title bar. The BPA script:

- Prompts the user to define the type of aggregation (this will present a list of measuring component types whose measuring component category is aggregator)
- Prompts the user to define the dimensional value(s), the desired aggregation time, and if this aggregation should be performed indefinitely or if it is a "one time"

## Base Product Aggregators

A number of standard aggregation processes are delivered as part of the base product. These provide a means of aggregating measurement data by postal code and service type for both physical meters and items.

| Aggregator Name | Description |
|---|---|
| Aggregator - Postal and Service Type | Aggregates measurement quantities for constituent measuring components from physical meter based on postal code and service type dimensions. |
| Item Aggregator - Postal and Service Type | Aggregates measurement quantities for constituent measuring component from items based on postal code and service type dimensions. |

### Aggregators Used With Analytics

Other aggregators shown below are delivered for use by Oracle Utilities Analytics and Oracle Utilities Analytics Visualization (OUAV). These aggregate measurement data (including quantities and counts) for constituent measuring components based on the following


Copyright © 2026, Oracle and/or its affiliates.

dimensions: Postal, City, Head-End, Device Type, Usage Calculation Group, Market and Service Provider, and Service Type.

| Aggregator Name | Description |
|---|---|
| Measurement Measured Quantity Aggregator<br>OUAV Subject Area: Meter - Aggregated Measurement Quantity | Aggregates measurement quantities and distributes the constituents' measurements across Value Identifiers based on condition codes. |
| Measurement Quality Count Aggregator<br>OUAV Subject Area: Meter - Aggregated Measurement Count | Aggregates measurement counts and distributes the constituents' counts across Value Identifiers based on condition codes. |
| Measurement Timeliness Count Aggregator<br>OUAV Subject Area: Meter - Aggregated Timeliness Count | Aggregates measurement counts that arrived on time or are late. |
| Measurement Timeliness Quantity Aggregator<br>OUAV Subject Area: Meter - Aggregated Timeliness Quantity | Aggregates measurement quantities that arrived on time or are late. |

See the Oracle Utilities Analytics Visualization User Guide for more information about the OUAV Subject Areas.

# Maintaining Aggregations

This portal is used to display and maintain Aggregations.

Refer to About Aggregations for additional functional information.

You can access the portal from the **Main > Totals and Trends > Consumption Statistics**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component**: defines the basic attributes of the aggregator measuring component.
- **Measuring Component Attributes:** displays attributes that describe the dimensions and criteria for the aggregation measuring component (applies to dynamic aggregation measuring components only)
- **Measurements — Interval**: displays aggregated measurement data for the selected aggregator measuring component.
- **Interval - Audit View**: allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement.
- **Final Values Overlay**: displays final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components.
- **Time of Use Overlay**: displays an overlay of the TOU periods on a final interval measurement along with totalized TOU consumption based on a user-defined time period and TOU map
- **Related Measuring Component Sets**: lists measuring components sets of which the aggregator measuring component is a member (applies to dynamic aggregation measuring components only)


Copyright © 2026, Oracle and/or its affiliates.

- **Usage Subscriptions Directly Linked to MC**: lists usage subscriptions that are directly linked to current measuring component

A few functions are available in many of these zones that allow you to interact with the measurement data. For more information of these functions refer to the Applying Measurement Functions section.

# Dynamic Aggregation

- Dynamic Aggregation Overview
- Understanding Dynamic Aggregation Execution
- Aggregation Measuring Components
- Viewing Dynamic Aggregation Results Using Oracle Utilities Analytics Visualization

## Dynamic Aggregation Overview

Standard aggregation processing uses specific algorithms configured to work with a specified set of dimensions. For example, the base package Service Type and Postal aggregation uses specific algorithms for dimension scanning and finding constituent measuring components. In contrast, dynamic aggregation uses dynamic queries for the dimension scanning, find constituents, and aggregation processes.

Dynamic aggregation can be used for analysis (to view aggregated data to understand load demand, for instance) as well as settlement processing performed by Oracle Utilities Market Settlements Management.

Dynamic aggregation results can be viewed using Oracle Utilities Analytics Visualization. See Viewing Dynamic Aggregation Results Using Oracle Utilities Analytics Visualization and the Oracle Utilities Analytics Visualization User Guide for more information.

## Understanding Dynamic Aggregation Execution

This section provides an overview of how dynamic aggregation processing is executed, including descriptions of the different types of dynamic aggregation processing and the different types of data that can be aggregated.

**Dynamic Aggregation Execution: Aggregation Groups and Aggregation Group Runs**

Aggregation processing is managed via aggregation groups. Aggregation Groups outline the sequence of aggregations that must occur for a given aggregation process. Each aggregation group includes a set of Measuring Component Sets, each of which has one or more related Aggregation Measuring Components.

When processing is initiated for an aggregation group, the system creates an Aggregation Group Run record to track the status and progress of the aggregation processing. This record is used to manage the aggregation process for each group and track which Measuring Component Sets should be executed next and when the aggregation is complete.

The following steps outlines the aggregation process:

1. The Dimension Scan and Aggregation monitor batch control (D1-DSAEM) initiates processing for all Aggregation Groups. This should be scheduled at least once per day.


Copyright © 2026, Oracle and/or its affiliates.

This monitor process runs all monitor algorithms on each aggregation group. The base package aggregation group has two monitor algorithms:

- Aggregation Group Run Maintenance (D1-AGRM)

2. The Aggregation Group Run Maintenance algorithm verifies if the Aggregation Group is ready to aggregate by looking at the processing timetable defined for the aggregation group and the last scheduled execution. If it is time for the next aggregation then the algorithm creates a pending Aggregation Group Run which will then be used to track the status of the Measuring Component Sets for the aggregation.
3. At the completion of the Dimension Scanner and Aggregation monitor batch process, the Submit Dimension Scanner Batch (D1-SUBDIMSCN) post-processing algorithm searches for any Aggregation Group Run entries that are in progress where the next Measuring Component Set is not manual. If at least one is found then it will initiate the Aggregation Group Dimension Scanner batch process (D1-AGDS).
4. The Aggregation Group Dimension Scanner batch process identifies the next Measuring Component Set for each Aggregation Group Run that is in progress. If that next Measuring Component Set is not manual then it will retrieve each Aggregation Measuring Component for the Measuring Component Set and execute the monitor algorithms for that Aggregation Measuring Component. These include an aggregation algorithm (specific to the type of data to be aggregated such as interval, billed quantities, etc.) and in some cases a value derivation algorithm.
5. For each Aggregation Measuring Component that is monitored by the Aggregation Group Dimension Scanner batch process, the aggregation process:
    a. Executes the appropriate Find Constituents algorithm to find the list of constituents that are to be aggregated
    b. Extracts the data for those constituents that should be aggregated
    c. Executes the appropriate aggregation algorithm against that data and store the results
    d. Executes validation algorithms if applicable
    e. Executes value derivation algorithms if applicable
6. The Aggregate Next Measuring Component Set (D1-AGGNXMCS) post-processing algorithm on the Aggregation Group Dimension Scanner batch process updates the Aggregation Group Run the batch information for each Measuring Component Set that was aggregated, and requests the Dimension Scan and Aggregation monitor batch process to run again.
7. This again executes the monitor algorithms for all the Aggregation Groups which will do one of several things:
    - If the Aggregation Group Run is in progress and the next Measuring Component Set is Automatic then nothing will happen.
    - If the Aggregation Group Run is in progress and the next Measuring Component Set is Manual it will create a To Do.
    - If the Aggregation Group Run does not have a next Measuring Component Set, it will set the Aggregation Group Run to Complete

> **Note**
>
> The Dimension Scanner algorithm will be executed again but since there will not be a pending Aggregation Group Run it will not do anything.


Copyright © 2026, Oracle and/or its affiliates.

8. Once all Measuring Component Sets are complete the Aggregation Group Run Maintenance algorithm marks the Aggregation Group Run as complete.

### Aggregation Scheduling

Refer to Understanding Aggregation Periods for basics about aggregation periods.

**Aggregation Scheduling** for Aggregation Groups can be based on either a Defined Lag, or a Processing Timetable.

- **Defined Lag** scheduling uses the following parameters:
  - **Aggregation Cut Off Time**: The end time for aggregation calculations performed for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Group. This is used to ensure a consistent end time for aggregation periods. This is especially useful when aggregating other aggregations.
  - **Aggregation Lag**: The number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the aggregation horizon that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.
  - **Aggregation Horizon**: The number of days in the aggregation period for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Group. This reflects the time period during which there's a potential change in measurement data for one or more of the measuring components associated with Aggregation Measuring Components. This together with the Aggregation Lag is used to determine the start and end dates of an aggregation period.
- **Processing Timetable** scheduling uses a Processing Timetable Type to drive aggregation schedule.

With dynamic aggregation, the aggregation period is consistent for all members of an Aggregation Group when an aggregation is performed. This is despite the fact that there may be differences between Measuring Component Sets and even Measuring Components within an Measuring Component Set as to how far the data has been aggregated through. In addition, individual Aggregation Measuring Components all use the same aggregation period when aggregation is performed.

### Anchor Time

Dynamic aggregation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on dynamic aggregation measuring component types and defines the start and end time of the business day for purposes of aggregation processing. When processing aggregation for dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When processing calculations for measuring components whose type specifies an **Anchor Time**, if the Start Time and End Time of the calculation period do NOT match the **Anchor Time** on the measuring component type, the Start and Stop Time are shifted to align with the closest time to the **Anchor Time** that aligns with the interval size of the measurement data being processed. The Start Time is shifted backwards (to an earlier time) and the End Time is shifted forwards (to a later time). In the case of hourly intervals with Start Times and End Times on the hour, no shift occurs. In the case of daily intervals, the shift is to the nearest Anchor Time. For example, in the Anchor Time is 07:00AM but the Start Time and End Time are both 12:00, the Start Time would be shifted to the previous 07:00AM, and the End Time would be shifted the next 07:00AM.

- Dynamic Aggregation Processing
- Types of Aggregations

## Dynamic Aggregation Processing

This section provides additional details around the specific types of processing used with dynamic aggregation.

Similar to standard aggregation, dynamic aggregation processing comprises three primary types of processes:

- **Dimension Scanning**: a process that searches for unique combinations of dimensions that are not yet represented by an Aggregation Measuring Component.
- **Find Constituents**: a process that finds all source data to be aggregated for a given Aggregation Measuring Component, based on the Aggregation Measuring Component's dimensions.
- **Aggregation**: a process that aggregates source data that corresponds to the Aggregation Measuring Component's dimensions.

Dynamic aggregation processing is driven by the following types of administrative data:

- **Data Sources** define the source of data to be aggregated, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or other sources of data.
- **Aggregation Measuring Component Types** define the most important properties of aggregation measuring components used to store aggregated data.
- **Aggregation Groups** define the ordering of a series of related aggregations and the schedule of aggregation.
- **Measuring Component Sets** define the dimensions and criteria by which aggregation will be performed.

Refer to Dynamic Aggregation Configuration Overview for more information about configuration of this data.

- Dynamic SQL
- Dimension Scanning
- Find Constituents
- Aggregation Calculation


Copyright © 2026, Oracle and/or its affiliates.

- Foundation and Composite Aggregation

## Dynamic SQL

Dynamic aggregation uses dynamic SQL queries to perform dimension scanning, find constituents, and aggregation processing. These dynamic queries are based on configuration of:

- Data Sources
- Aggregation Measuring Component Types
- Measuring Component Sets

**The Source of Dynamic SQL**

Data sources provide template SQL that is used as the basis for dynamic queries. This template SQL is extended by additional SQL generated from the data source configuration as well as configuration defined for measuring component sets, measuring components, and measuring component types that reference the data source. The ways in which the template SQL is extended is based on the type of process:

- **Dimension Scanning**: The data source template SQL is extended by the criteria, dimensions, and individually managed items configured on the measuring component sets being processed.
- **Find Constituents**: The data source template SQL is extended by attributes defined on the aggregation measuring components being processed.
- **Aggregation**: The data source template SQL is extended by the value identifiers defined on the measuring component type of the aggregation measuring components being processed.

## Dimension Scanning

The dimension scanning process analyzes master data to identify unique dimension combinations and create an aggregation measuring component for each. Several objects play a role in this process:

- The Aggregation Group holds the monitor algorithm that performs the scanning
- The Data Source defines the template SQL for identifying dimensions
- Measuring Component Sets define the dimensions and criteria for identifying the dimension combinations

See Dynamic SQL for more about the role of data sources and measuring component sets in generating dynamic SQL queries for dimension scanning.

This process is executed when aggregation is initiated for an Aggregation Group. This process should be run on a frequent basis (daily or more frequent) and should be executed prior to aggregation processing being run. Each time it is executed the dimension scanning process takes the following high level steps:

1. Identifies all Measuring Component Sets for the Aggregation Group that are not imported from an external source
2. Identifies all yet to be represented dimension combinations
3. Creates an Aggregation Measuring Component based on the Measuring Component Set's Measuring Component Type for each dimension combination identified in step 2.


Copyright © 2026, Oracle and/or its affiliates.

4. Adds Attributes for each Aggregation Measuring Component created to describe the dimensions and criteria for that Measuring Component
5. Adds an entry in the Measuring Component Set participants table for each Aggregation Measuring Component created

This processing is performed by a monitor algorithm (Aggregation Group Dimension Scanner: D1-AGDS) on the Finalized status of the Aggregation Group business object.

## Find Constituents

The Find Constituents process finds all source data to be aggregated for a given aggregation measuring component, based on the Aggregation Measuring Component's dimensions. The queries used for this process are based on data source template SQL extended by attributes defined on the Aggregation Measuring Components being processed.

The Find Constituents process is initiated by the aggregation algorithm on the Aggregation Measuring Component (each type of Aggregation Measuring Component — interval, item, billed quantities — has its own aggregation algorithm), and is performed by algorithms defined for the Find Constituents system event on the Aggregation Measuring Component business object.

This process differs slightly when finding constituents for interval and scalar measurements stored on measuring components, item data, and billed quantities.

### Measuring Components

The Find Constituents process for measuring components uses the Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm which retrieves data based on the following:

- Attributes on the Aggregation Measuring Component based on the Measuring Component Set configuration
- Measuring Component Types on the Data Source

Measuring component constituents are inserted into a global temporary table (GTT1) with the following information:

- Measuring Component ID
- Horizon Start Date/Time
- Horizon End Date/Time
- Time Zone

> **Note**
>
> The aggregation period for each is adjusted by the source Measuring Component's installation history.

### Items

The Find Constituents process for items uses the Find Constituents for Items (D1-FICONITEM) algorithm which retrieves data based on the following:

- Attributes on the Measuring Component based on the Measuring Component Set configuration


Copyright © 2026, Oracle and/or its affiliates.

- Item Types / Service Point Types on the Data Source (for badged vs. unbadged items, respectively)

Item constituents are inserted into a global temporary table (GTT3) with the following information:

- Maintenance Object (Device or Service Point)
- Device Type: Item Type
- Item Count
- Horizon Start Date/Time
- Horizon End Date/Time
- Time Zone

> **Note**
>
> The aggregation period for each is adjusted by the source Item's installation history.

**Billed Quantities**

The Find Constituents process for billed quantities uses the Find Constituents for Billing Data (D1-FINDCONSB) algorithm which retrieves data based on the following:

- Attributes on the Aggregation Measuring Component based on the Measuring Component Set configuration
- Usage Subscription Types on the Data Source

Billed quantity constituents are inserted into a global temporary table (GTT5) with the following information:

- Usage Subscription ID
- Usage ID
- Horizon Start Date/Time
- Horizon End Date/Time
- Time Zone

> **Note**
>
> The aggregation period for each is adjusted by the source Usage Subscription's installation history.

## Aggregation Calculation

The aggregation calculations process aggregates source data that correspond to the Aggregation Measuring Component's dimensions. The queries used for this process are based on data source template SQL extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Components being processed.

The aggregation calculation process is initiated by Aggregation by MC Set Monitor batch process (D1-AMCS), and is performed by monitor algorithms defined on the Active state of the


Copyright © 2026, Oracle and/or its affiliates.

Aggregation Measuring Component business object. The algorithms used by each type of Aggregation Measuring Component are as follows:

| Aggregation Measuring Component Business Object | Aggregation Algorithm |
|---|---|
| Dynamic Aggregator - Billed Quantities (D1-AggBilledServiceQuantities) | Aggregate Billed Service Quantities (D1_AGGBSQ) |
| Dynamic Aggregator - Composite (D1-AggregatorComposite) | Aggregate Interval Measurements (D1_AGGINT) |
| Dynamic Aggregator - Customer Quantities (D1-AggregatorCustomerQuantity) | Aggregate Customer Specific Quantities (D1_AGGCSQ) |
| Dynamic Aggregator - Interval (D1-AggregatorInterval) | Aggregate Interval Measurements (D1_AGGINT) |
| Dynamic Aggregator - Items (D1-AggregatorItems) | Aggregate Item Data (D1_AGGITEM) |
| Dynamic Aggregator - Scalar (D1-AggregatorScalar) | Aggregate Scalar (D1_AGGSCLR) |

See Types of Aggregation for more information about aggregation calculations for each type.

### Other Calculations

In addition to basic aggregation processing, other calculations can be performed on aggregated measurements as needed. For instance, a common step in aggregating date for use in market settlements is applying a loss factor to aggregated data to reflect system losses (line loss, etc.) between generation-level quantities and end consumption quantities.

These types of calculations can be performed by value derivation algorithms. These are algorithms based on the Derive Aggregated Quantity Using a Formula (D1-DERAGGQTY) algorithm type which apply a configured formula to aggregated measurement values. For instance, to apply a loss factor, an algorithm can retrieve effective-dated factor values for a loss factor and multiply each aggregated measurement value by the appropriate factor value.

The Execute Aggregation Value Derivation (D1-EAGGVALD) algorithm on Aggregation Measuring Components executes any value derivation algorithms specified for Value Identifiers on the Aggregation Measuring Component Type.

### Aggregation Calculation and Anchor Time

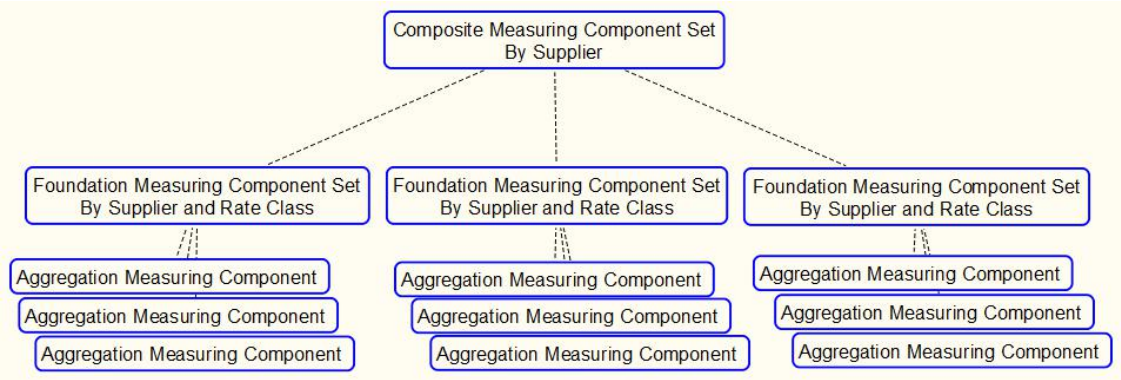
Dynamic aggregation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on dynamic aggregation measuring component types and defines the start and end time of the business day for purposes of aggregation processing. When processing aggregation for dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.

## Foundation and Composite Aggregation

Dynamic aggregation supports two “levels” of aggregation. Aggregating source data, from measuring components, items, or billed quantities is referred to as “Foundation” aggregation. Aggregation of aggregated data is known as “Composite” aggregation. Composite aggregation provides the ability to aggregate data from a set of other aggregations.

The diagram below illustrates a simple example of the relationship between foundation and composite measuring component sets. Each of the three foundation measuring component sets has a number of aggregation measuring component participants that aggregate data based on a set of dimensions and a particular supplier and rate class. Those in turn are participants in the composite measuring component set that aggregates based on the supplier (including data for all rate classes).


Copyright © 2026, Oracle and/or its affiliates.




A common scenario where composite aggregation would be used is to consolidate data across several foundation aggregation measuring component sets that pulled data from a set of data sources ordered by precedence such that the best data was sourced when available and when not the aggregations fell back to a lesser quality data source:

- Foundation: Interval measurements from usage subscriptions and / or service points
- Foundation: Scalar measurement from usage subscriptions and / or service points
- Foundation: Interval measurements from estimations (backcasted using actual weather)
- Composite: Aggregation by supplier

In this example the goal of the first aggregation would be to retrieve as much interval data as possible where that interval data covered the entire period being settled for. The second aggregation would take scalar data for those instances where interval data was not available and apply a profile to produce the appropriate intervals. The third aggregation would fill in any remaining gaps that were not covered by interval or scalar data with estimations based various forecasting models.

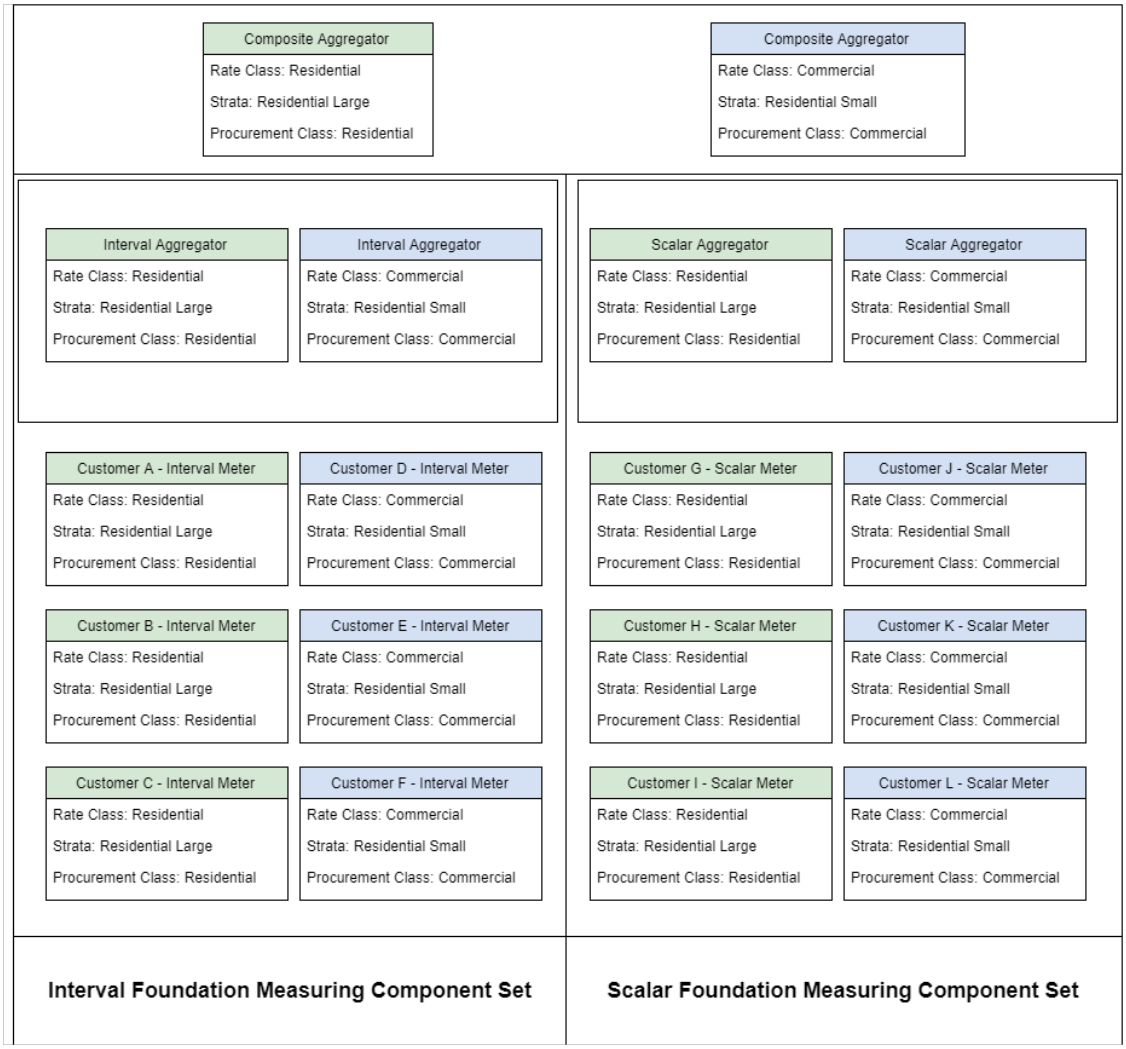
At the completion of those first three aggregations there would be the full resolution of data for the customer base. However, the values would be spread out across many aggregation measuring components tied to those measuring component sets. This is where the composite aggregation functionality comes into play. The composite aggregation measuring component would reference the first three foundation aggregations and sum together the data from each aggregation into a single instance that represents the total usage for the period being settled.

Let’s look at a slightly more involved example. The diagram below illustrates a set of foundation aggregations based on:

- Rate Class
- Strata
- Procurement Class

This shows how the physical measuring component data (designated by Customer A, Customer B, etc.) will be aggregated into separate aggregation measuring components based on source of data (interval or scalar). Then it shows how those aggregations are then aggregated to produce a total number that represents all 3 aggregation sources merged together.


Copyright © 2026, Oracle and/or its affiliates.




# Types of Aggregations

This section describes the processing used by different types of aggregations, including:

- Aggregating Measurement Data (Interval and Scalar)
- Aggregating Item Data
- Aggregating Billed Service Quantities
- Aggregating Aggregated Data (Composite Aggregation)


- Aggregating Measurement Data
- Aggregating Item Data
- Aggregating Billed Quantities
- Aggregating Aggregated Data


## Aggregating Measurement Data

Aggregating measurement data involves identifying constituent measuring components for an Aggregation Measuring Component, identifying the measurement data that falls within the



Copyright © 2026, Oracle and/or its affiliates.

aggregation period, and aggregating that data. The process varies slightly between interval and scalar measuring components.

### Interval Measuring Components

Aggregation of interval measurements is handled by the Aggregate Interval Measurements (D1_AGGINT) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm and populates the GTT1 global temporary table (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT1 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT1, the algorithm inserts its measurements for the aggregation period into the GTT2 global temporary table. If the SPI and UOM are different from that of the Aggregation Measuring Component, the measurements are converted to a common SPI using Axis Conversion.
- It performs aggregation calculations on the measurement data in the GTT2 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.
- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

### Scalar Measuring Components

Aggregation of scalar measurements is handled by the Aggregate Scalar Measurements (D1_AGGSCL) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm and populates the GTT1 global temporary table (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT1 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT1, the algorithm inserts its measurements for the aggregation period into the GTT2 global temporary table. If the UOM is different from that of the Aggregation Measuring Component, the measurement is converted to a common UOM before being inserted into GTT1.
- It performs aggregation calculations on the measurement data in the GTT2 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It applies a profile to the aggregated scalar values to create interval measurements with the same SPI as the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.


Copyright © 2026, Oracle and/or its affiliates.

- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

## Aggregating Item Data

Aggregating item data involves identifying constituent items for an Aggregation Measuring Component, calculate item measurement data for the aggregation period, and aggregating that data.

Aggregation of interval measurements is handled by the Aggregate Item Data (D1_AGGITEM) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Items (D1-FICONITEM) algorithm and populates the GTT3 global temporary table with groups of device types (for badged items) and/or service point types (for unbadged items) (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT3 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT3, the algorithm calculates measurements for the aggregation period and inserts them into the GTT4 global temporary table by applying the item counts for each group to the item interval (profile) data. If the SPI and UOM are different from that of the Aggregation Measuring Component, the measurements are converted to a common SPI using Axis Conversion.
- It performs aggregation calculations on the measurement data in the GTT4 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.
- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

## Aggregating Billed Quantities

Aggregating billed service quantities involves identifying constituent usage transactions for an Aggregation Measuring Component, retrieving the service quantity data that falls within the aggregation period, and aggregating that data.

Aggregation of billed quantities is handled by the Aggregate Billed Service Quantities (D1_AGGBSQ) algorithm. The algorithm performs the following steps:

- It calls the Find Constituents for Billing Data (D1-FINDCONSB) algorithm and populates the GTT5 global temporary table (see Find Constituents for more details about this process).
- It deletes any constituents from the GTT5 table that do not have 100% coverage for the dates in the aggregation period.
- For each remaining constituent in GTT5, the algorithm retrieves billing data for the aggregation period (based on the service quantity configuration on the data source) and inserts it into the GTT6 global temporary table. If the UOM is different from that of the Aggregation Measuring Component, the service quantities are converted to a common UOM.


Copyright © 2026, Oracle and/or its affiliates.

- It performs aggregation calculations on the measurement data in the GTT6 table, using data source template SQL, extended by the value identifiers defined on the measuring component type of the Aggregation Measuring Component.
- It applies a profile to the aggregated values to create interval measurements with the same SPI as the Aggregation Measuring Component.
- It either inserts or updates the results to the appropriate measurement table (the standard Measurement table, or the Aggregation Measurement table)
- If there is no data for any dates in the aggregation period and previously aggregated measurements exist, those measurements are deleted to indicate the absence of data.
- It updates the **Most Recent Measurement Date/Time** and **Latest Measurement Date/Time** values on the Aggregation Measuring Component.

### Aggregating Aggregated Data

Aggregating aggregated data, also known as composite aggregation, involves identifying constituent aggregation measuring components for a composite Aggregation Measuring Component, identifying the measurement data that falls within the aggregation period, and aggregating that data.

Composite aggregation is very similar to aggregating interval measurement data, and uses the Aggregate Interval Measurements (D1_AGGINT) Find Constituents for Dynamic Aggregation (D1-FINDCONST) algorithm.

## Aggregation Measuring Components


- About Aggregation Measuring Components
- Maintaining Aggregation Measuring Components


### About Aggregation Measuring Components

Aggregation Measuring Components are measuring components used to store aggregated data. Aggregation measuring components do not use initial measurements. Aggregated measurements are stored in the Measurement or Aggregation Measurement table (based on the Measurement Business Object defined for the aggregation measuring component's type).

Aggregation Measuring Components are defined by the following:

- **Initiating Measuring Component Set**: The measuring component set that triggered creation of the aggregation measuring component via the Dimension Scanning process (see Dimension Scanning for more information).
- **Processing Information**: Details of aggregation processing for the aggregation measuring component, including the date and time through which aggregation has been performed.
- **VEE Groups**: The VEE groups used when performing validation, editing, and estimation for the aggregation measuring component.
- **Measuring Component Attributes**: Attributes that describe the dimensions and criteria for the aggregation measuring component (identified by the Dimension Scanning process)

The base package includes the following aggregation measuring component business objects for use with dynamic aggregation:



Copyright © 2026, Oracle and/or its affiliates.

- Aggregator Interval (D1–AggregatorInterval): Used to aggregate interval measurements from interval measuring components
- Aggregator Scalar (D1–AggregatorScalar): Used to aggregate scalar measurements from scalar measuring components
- Aggregator Items Measuring Component (D1–AggregatorItems): Used to aggregate item data
- Aggregator Billed Service Quantities (D1–AggBilledServiceQuantities): Used to aggregate billed service quantities from usage transactions
- Aggregator Customer Quantities (D1–AggregatorCustomerQuantity): Used to aggregate specific types of customer quantities such as Peak Load Contribution (PLC) and usage factors
- Aggregator Composite (D1–AggregatorComposite): Used to aggregate aggregated data from aggregation measuring components (composite aggregation)

These business objects define the Find Constituents and Aggregation algorithms used for aggregation calculations.

## Maintaining Aggregation Measuring Components

You use the **Totals and Trends View** portal to maintain aggregation measuring components.

Refer to About Aggregation Measuring Components for additional information.

You can access this portal by selecting an aggregation measuring component from the **Measuring Component Set Participants** zone in the **Measuring Component Set** portal. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

This portal includes the following zones:

- **Measuring Component**: This zone displays the basic attributes of the aggregation measuring component
- **Measuring Component Attributes:** This zone displays attributes that describe the dimensions and criteria for the aggregation measuring component
- **Related Measuring Component Sets**: This zone displays the measuring component sets related to the aggregation measuring component
- **Final Values Overlay**: This zone graphs final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components. The zone also permits overlaying data from the same measuring component for different time periods, as well as data from measuring components that measuring different quantities, such as temperature.
- **Interval - Audit View**: This zone allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the curve and see how the measurements looked at different points in time.
- **Measurements - Interval**: This zone displays measurement details for the aggregation measuring component
- **Usage Subscriptions Directly Linked to MC**: This zone displays usage subscriptions that are directly linked to the aggregation measuring component


Copyright © 2026, Oracle and/or its affiliates.

- **Time of Use Overlay**: This zone displays an overlay of the TOU periods on measurement data along with totalized TOU consumption based on a user-defined time period and TOU map

# Viewing Dynamic Aggregation Results Using Oracle Utilities Analytics Visualization

This topic describes objects used to support viewing dynamic aggregation results using Oracle Utilities Analytics Visualization. Refer to the Oracle Utilities Analytics Visualization User Guide for more information about using Oracle Utilities Analytics Visualization.

### Dynamic Aggregation Datasets

The Oracle Utilities Analytics Visualization demonstration database includes predefined datasets that can be used to view dynamic aggregation results using Oracle Utilities Analytics Visualization.

The “Meter — Usage by Network Location” and “Meter — Usage by Customer Class” sample datasets allow access to dynamic aggregation results. Users access these from the **Datasets** tab in the Oracle Business Intelligence user interface.

These datasets use SQL to retrieve data from the dynamic aggregation results.

### Meter — Usage by Network Location

The “Meter — Usage by Network Location” sample dataset uses the following SQL, based on dimensions including City, Postal Code, Commodity, Facility (Transformer), Feeder, and Substation for a given date/time range.

```
select VW.MEASR_COMP_SET_CD MC_Set,
       VW.AGG_MEASR_COMP_ID MC,
       VW.AGG_ATTR1 City,
       VW.AGG_ATTR2 Postal,
       decode(VW.AGG_ATTR3, 'D1CD', 'Commodity', VW.AGG_ATTR3) Relation,
       VW.AGG_ATTR4 Facility_ID,
       nvl(FI.ID_VALUE, 'N/A') Transformer,  /* Facility Identifier */
       nvl(NL.FEEDER, 'N/A') FEEDER,
       nvl(NL.SUBST, 'N/A') SUBST,
       VW.MSRMT_DTTM Date_Time,
       VW.AGG_VAL Usage,
       VW.AGG_VAL1 Load
 from     D1_BI_DYN_AGG_VW VW
left join (select * from
            (select distinct L.FACILITY_ID, NETWORK_NODE,
NETWORK_NODE_TYPE_FLG
              from D1_NW_NODE N
             inner join D1_NW_LOC L on L.NETWORK_LOCATION_ID =
N.NETWORK_LOCATION_ID)
        pivot (max(NETWORK_NODE) for NETWORK_NODE_TYPE_FLG
            in ('D1-Feeder' as FEEDER, 'D1-Substation' as SUBST))) NL on
NL.FACILITY_ID = VW.AGG_ATTR4
left join D1_FACILITY_IDENTIFIER FI on FI.FACILITY_ID = VW.AGG_ATTR4 and
FI.FACILITY_ID_TYPE_FLG = 'D1EI'
where MEASR_COMP_SET_CD IN('DA-E-CONS-TRANS')
```


Copyright © 2026, Oracle and/or its affiliates.

### Meter — Usage by Customer Class

The “Meter — Usage by Customer Class” sample dataset uses the following SQL, based on dimensions including City, Postal Code, Rate Class, and Profile Class for a given date/time range.

```
select VW.MEASR_COMP_SET_CD MC_Set,
       VW.AGG_MEASR_COMP_ID MC,
       VW.AGG_ATTR1 City,
       VW.AGG_ATTR2 Postal,
       VW.AGG_ATTR3 Rate_Class_CD,
       decode(VW.AGG_ATTR3, 'E-TOU', 'Electric TOU',
                            'ER-BASIC', 'Electric Residential Basic',
                            'ER-DMINU', 'Electric Residential Stepped',
                            VW.AGG_ATTR3) Rate_Class,
       nvl(RS.DESCR, VW.AGG_ATTR3) Rate_Descr,
       VW.AGG_ATTR4 Profile_Class_CD,
       nvl(PR.DESCR, VW.AGG_ATTR4) Profile_Class,
       VW.MSRMT_DTTM Date_Time,
       VW.AGG_VAL Usage,
       VW.AGG_VAL1 Load
     from D1_BI_DYN_AGG_VW VW
left join CI_RS_L          RS  on RS.RS_CD = VW.AGG_ATTR3
                               and RS.LANGUAGE_CD = 'ENG'
left join CI_CHAR_VAL_L    PR  on PR.CHAR_TYPE_CD = 'DM_PROCL'
                               and trim(PR.CHAR_VAL) = VW.AGG_ATTR4
                               and PR.LANGUAGE_CD = 'ENG'
where MEASR_COMP_SET_CD IN('DA-E-CONS-PROF')
```

The SQL used by these datasets must be updated and adjusted based on the dimensions used in your dynamic aggregation configuration, and the Measuring Component Set (MEASR_COMP_SET_CD).

### Dynamic Aggregation View — Dynamic Aggregated Measurements

The application includes a “Dynamic Aggregated Measurements” view which can be used when viewing dynamic aggregation results using the “Meter — Usage by Network Location” and “Meter — Usage by Customer Class” sample datasets via Oracle Utilities Analytics Visualization. Details for this view are provided below.

#### Properties

| | |
|---|---|
| **Name** | D1_BI_DYN_AGG_VW |
| **Description** | Dynamic Aggregated Measurements |
| **Date/Time Data Type** | Local Legal Time |
| **Upgrade** | Keep |
| **Key Validation** | Always Check Uniqueness |

#### Columns

Columns without extra specification are standard existing fields.


Copyright © 2026, Oracle and/or its affiliates.

| Seq | Column | Nullable | Key | Comment |
|---|---|---|---|---|
| 1 | MEASR_COMP_SET_CD | No | PK, FK | D1_MEASR_COMP_SET |
| 2 | AGG_MEASR_COMP_ID | No | PK | |
| 3 | MSRMT_DTTM | Yes | PK | |
| 4 | AGG_ATTR1 | Yes | | |
| 5 | ... | Yes | | |
| 6 | AGG_ATTR20 | Yes | | |
| 7 | AGG_VAL | Yes | | |
| 8 | AGG_VAL1 | Yes | | |
| 9 | ... | Yes | | |
| 10 | AGG_VAL10 | Yes | | |

**Constraints**

| ID | Column | Flag | Table | Referred |
|---|---|---|---|---|
| | MEASR_COMP_SET_CD | PK | | |
| | AGG_MEASR_COMP_ID | PK | | |
| | MSRMT_DTTM | PK | | |
| | MEASR_COMP_SET_CD | FK | D1_MEASR_COMP_SET | MEASR_COMP_SET_CD |

# Master Data Sync

The synchronization of master data into Oracle Utilities Meter Data Management is covered in detail in the Integrations section of the *Oracle Utilities Meter Solution Administrative User Guide*.

# Outage Storm Mode

- About Outage Storm Mode
- Reviewing Widespread Outages

## About Outage Storm Mode

Unfortunately, utilities and their customers are often subject to natural disasters that cause disruptions to AMI networks. These outage events can be related to hurricanes, tornadoes, ice storms, earthquakes, tsunamis, etc. During these severe outages, AMI systems do not always provide outage and restoration events from the devices. Without these outage and restoration


Copyright © 2026, Oracle and/or its affiliates.

events, Oracle Utilities Meter Data Management processes could estimate measurements that would be billed by the CIS. This will lead to unhappy customers, potentially bad press, and lots of manual cleanup work after the fact.

While this is a problem for major outage events, this can also occur with smaller outages (smaller blackouts, planned outages, etc.). This can potentially become an issue anytime the AMI network has a significant number of devices failing to communicate.

This module includes outage processing to improve estimation accuracy during major outages. The system will automatically stop estimations when the number of meter readings coming is less than a user defined tolerance. This tolerance is based on a significant drop in the read percentage for all devices in a similar postal code, service type, and Head End. Estimations will automatically resume when the device returns to normal operation. This can occur through a number of actions: read percentage for the surrounding area returns to normal, regular data is received from the device, or a power restoration device event is received from the device.

Oracle Utilities Meter Data Management also includes the ability to manually override "storm mode". This override will allow estimations to resume even if the outage hasn't been fully resolved. This will be useful if network fixes take too long and the utility has decided that customers should be billed on an estimate after a few days pass.

> **Note**
>
> For information on configuration, reference the related section for Outage Storm Mode in the Administrative User Guide.

The diagram below illustrates the primary flow for Outage Storm Mode processing. The steps include:

1. The **Dimension Scanner** batch (D1-ADS) monitors the **Meter Communication Tracking Dimension Scanner** business object. New Aggregator Measuring Components will be created for every applicable Service Type, Postal Code, and Head End in the system.
2. The **Aggregation Monitor** batch (D2-AGG) runs to execute the logic for the **Meter Communication Tracking Aggregator** Measuring Component. This logic will aggregate all meters for the defined Service Type, Postal Code, and Head End. Once the read percentage is found for that day then a measurement will be created for the Aggregator Measuring Component.
3. The **Widespread Outage** batch (D2-WSO) runs to determine if any of the "Meter Communication Tracking Aggregator" Measuring Components have dropped below the defined threshold on the MDM Master Configuration. If it has dropped enough, then an **Estimation Suppression** Activity is created for each of the affected Service Points and the "Meter Communication Tracking Aggregator" Measuring Component is set to an "Active Outage" indicator.
4. If the **Device Periodic Estimation** batch (D1-DVCPE) runs to generate estimates while an Estimation Suppression Activity is active for a Service Point, then the Service Point will either be skipped for estimation or will have zero consumption filled for all data with a condition code of "No Read - Outage Fill".
5. There are a number of methods available to resume normal operation and close out the Estimation Suppression Activity. These include the following:
    a. The read percentage reaches a "normal level" again as defined in the MDM Master Configuration. This will be detected by the **Widespread Outage** batch (D2-WSO).
    b. When non-estimated data is received from a device, this will close out an open Estimation Suppression Activity.

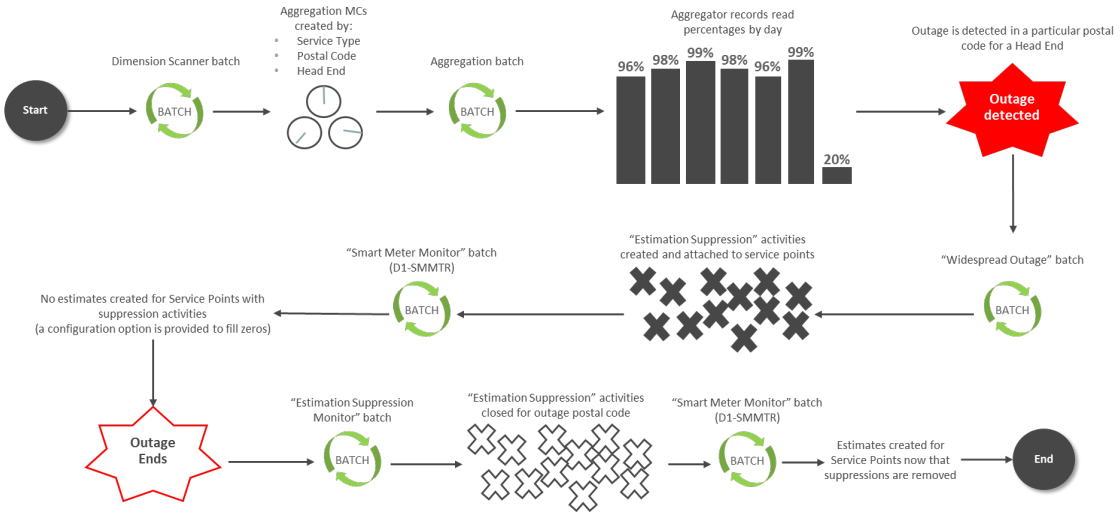

Copyright © 2026, Oracle and/or its affiliates.

c. If the **Standard Event Name** Extendable Lookup is configured with the End Estimation Suppression algorithm (D2-EN-ESTSUP), then any Power Up event this is configured for will close out an open Estimation Suppression Activity.

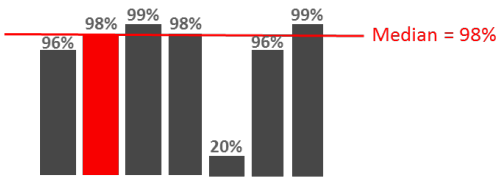
d. Estimation Suppression Activities can also be closed manually be a user by navigating to the Activity, using the "Edit" button", and add an End Date.

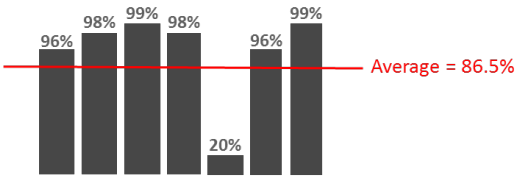
6. The **Estimation Suppression Monitor** batch (D2-ESTSU) runs for all Estimation Suppression Activities to determine if the related "Meter Communication Tracking Aggregator" Measuring Component is no longer in a state of "Active Outage". If the widespread outage has ended, then the Estimation Suppression activity will be set with an end date.

7. The next time the **Device Periodic Estimation** batch (D1-DVCPE) runs it will generate estimated data as normally configured.




In order to determine if an area is experiencing a widespread outage, the median read percentage of prior days is used as the basis for comparison. The median was chosen instead of the average to help reduce the impact of outlier values. This is illustrated in the example below.




# Reviewing Widespread Outages

Anytime a **Meter Communication Tracking Aggregator** Measuring Component is marked with an indicator of "Active Outage", a To Do will be generated for user review. The base To Do Type for this is "Widespread Outage" (D2-WSOTD). This To Do is meant to serve as a warning to users that one of the areas in the service territory is experiencing a widespread outage.


Copyright © 2026, Oracle and/or its affiliates.

If there is a desire to continue processing as normal and not have estimation suppressed, then you can override the Meter Communication Tracking Aggregator by hitting the "Force Complete Outage" button while reviewing the To Do. This will mark the Widespread Outage Indicator as "User Override" and will allow estimation to continue normally.

# Reports


- About Reports
- Running Oracle Utilities Meter Data Management Reports


## About Reports

Reports include operational reports related to device events, validation exceptions, and other operational data. Reports can also be based on devices, measuring components, Service Points, service providers, and other data.

Oracle Utilities Meter Data Management reports are implemented using Oracle Business Intelligence Publisher (BI Publisher), and are administered via Oracle Business Intelligence Publisher and the Report functionality of the Oracle Utilities Application Framework.

Refer to the Oracle Business Intelligence Publisher documentation for more information about installing Oracle Business Intelligence Publisher and creating and defining reports. For additional details on how reports are defined and administered see Reporting and Monitoring Tools in the Oracle Utilities Application Framework *Administrative User Guide*.

## Running Oracle Utilities Meter Data Management Reports

This portal is used to submit a new instance of a Report.

Refer to About Reports for additional functional information.

You can access the portal from the **Main > Tools > Report Submission**.

First, the appropriate Report code must be selected. The Reports provided out-of-the-box are:

- **Device Events Report**: this report generates a pie graph depicting the percentage of Device Events for each Device Event Type for the specific search criteria.
- **Device VEE Exceptions Report**: this report generates a pie graph depicting the percentage of VEE Exception for Exception Type and Market for the specific search criteria.

Next, enter the parameter values for the report as appropriate and submit the report.

# Dashboards


- About Dashboards
- Using the Meter Operational Dashboard
- Using the Service Order Operational Dashboard




Copyright © 2026, Oracle and/or its affiliates.

- Using the Service Order Trends Dashboard

# About Dashboards

Dashboards provide a broad view of numerous metrics used with Oracle Utilities Meter Data Management, Oracle Utilities Smart Grid Gateway, and other products. The data presented on each dashboard is tailored to provide operational as well as analytical information for management. At times, drill down options are provided to retrieve more detailed information about the metrics provided.

For information on dashboard configuration, reference the related section for Dashboards in the *Oracle Utilities Meter Solution Administrative User Guide*.

# Using the Meter Operational Dashboard

This portal is used to display the Meter Operational Dashboard.

You can access the portal by selecting **Main , Dashboards, Meter Operational Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

The **Meter Operational Dashboard** contains three tabs:

- Operations
- Batch Performance
- System Health Check

### Using the Operations Tab

The **Operations** tab displays the following zones:

- **IMD Loading Trend**: displays the count of initial measurement data that's loaded each day by Head End system.
- **Device Event Loading Trend**: displays the count of device event data that's loaded each day by Head End system.
- **Incomplete IMDs by Status**: displays a set of counts for initial measurement data by status.
- **Incomplete IMDs Trend**: displays a set of counts for initial measurement data for each day by status
- **VEE To Dos by Exception Type**: displays a set of counts for VEE Exceptions by Exception Type that also still have a To Do that's incomplete.
- **VEE To Dos Trend**: displays a set of counts for VEE Exceptions by Exception Type for each day that also still have a To Do that's incomplete.
- **Usage Transactions by Status**: displays a set of counts for usage transactions by status.
- **Usage Transactions Trend**: displays a set of counts for usage transactions for each day by status.
- **Usage Transaction To Dos by Exception**: displays a set of counts for Usage Transaction Exceptions by Exception Type that also still have a To Do that's incomplete.
- **Usage Transaction To Dos Trend**: displays a set of counts for Usage Transaction Exceptions by Exception Type for each day that also still have a To Do that's incomplete.


Copyright © 2026, Oracle and/or its affiliates.

Click **Configure** to navigate to the MDM Operational Dashboard Configuration master configuration. See **Configuring the Meter Operational Dashboard** in the *Administrative User Guide* for information regarding configuration of the dashboard.

**Filtering by Division**

Data displayed in the zones on the **Operations** tab (except the **IMD Loading Trend** and **Device Event Loading Trend** zones) can be filtered by Division. To enable filtering by Division set the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration to "Yes". If this is set to “No” the zones display data for all divisions.

When filtering is enabled, the **Division** drop-down list allows the user to select the Division for which they wish to view data. Available divisions are based on the divisions assigned to the user in the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. The division marked “Use As Primary” is the default.

- Using the Batch Performance Tab
- Using the System Health Check Tab

# Using the Batch Performance Tab

The **Batch Performance** tab shows a series of zones related to any defined batch-based Performance Targets

Click **Configure** to navigate to the **Performance Target** portal. See Measuring Performance in the *Administrative User Guide* for information.

# Using the System Health Check Tab

The **System Health Check** tab includes a set of zones displaying data related to initial measurement processing and quality. These include:

- IMD Processing Trend
- IMD Processing Detail
- IMD Quality Trend
- IMD Quality Detail
- IMD Control Seeder Overview
- IMD Control Seeder Trend

Click **Configure** to navigate to the System Health Check Dashboard Configuration master configuration. See **Configuring the System Health Check Dashboard** in the *Administrative User Guide* for information regarding configuration of the dashboard.

**IMD Processing Trend**

The **IMD Processing Trend** zone displays the number of initial measurements and calculated numbers of constituent measurements for each day in the selected time period. The x-axis displays individual dates during the selected period. The left y-axis shows the average and maximum number of measurements per initial measurement. The right y-axis shows the number of initial measurements. The graph shows the following:

- **IMD Records**: The number of initial measurements processed on each day in the selected time period.


Copyright © 2026, Oracle and/or its affiliates.

- **Average Measurements**: The average number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.
- **Maximum Measurements**: The maximum number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the largest number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.

You can filter the data displayed using the following controls:

- **Period View**: Specifies the range of data displayed. Options include Thirty Days, Seven Days, Three Days, or One Day. The default is "Seven Days".
- **Division**: Specifies the division of the initial measurements displayed in the graph. The default is "Show All".
- **Interval/Scalar**: Specifies whether the graph should display only interval initial measurements, scalar initial measurements, or both. The default is "Show All".
- **Business Object**: Specifies the initial measurement business object(s) related to the initial measurements displayed in the graph. The default is "Show All".
- **Status**: Specifies the status (Pending, VEE Ready, Exception, Error, Finalized, and so on) of the initial measurements displayed in the graph. The default is "Show All".
- **Data Source**: Specifies the data source (if any) of the initial measurements displayed in the graph. The default is "Show All".

Click on the labels on the legend below the graph to hide/show specific metrics.

This zone displays data from statistics snapshots based on the IMD Processing Snapshot - Hourly (D1-IMDProcessingHourly) business object. The **Snapshot Date/Time** drop-down list (in the lower right corner of the zone) lists all currently available snapshots based on this business object. Filtering options are based on data available in the snapshot. For example, if all of the initial measurements processed for a specific snapshot are all interval initial measurements, the "scalar" option wouldn't appear in the **Interval/Scalar** drop-down list.

Other metrics, including File Count (the number of payload files processed) and Total Measurements, Average Age, and Maximum Age can be viewed from the statistics snapshot.

Click **Capture** (next to **Snapshot Date/Time** in the lower right corner of the zone) to navigate to the **Statistics Control** portal where you can view a list of snapshots for the statistics control, view details of a specific snapshot, and create a new snapshot if needed.

#### IMD Processing Detail

The **IMD Processing Detail** zone displays details from the most recent snapshot based on the IMD Processing Snapshot - Hourly (D1-IMDProcessingHourly) business object. By default this zone displays a tabular view of the data from the snapshot, with each row in the table representing a unique combination of business object, interval/scalar flag, and status. This table includes the following:

- **Date Time**: The date and time on which the initial measurements captured in the row were processed
- **Business Object**: The business object of the initial measurements captured in the row


Copyright © 2026, Oracle and/or its affiliates.

- **Interval/Scalar**: A flag designating if the data captured in the row is interval (D1IN) or scalar (D1SC)
- **Status**: The status of the initial measurements captured in the row
- **Description**: A description of the initial measurements captured in the row
- **IMD Records**: The number of initial measurements processed for the unique combination of business object, interval/scalar, and status captured in the row
- **Average Measurements**: The average number of measurements (Total Measurements divided by IMD Records) per initial measurement captured in the row
- **Maximum Measurements**: The maximum number of measurements per initial measurement captured in the row
- **Total Measurements**: The total number of measurements values in the initial measurements captured in the row

To view a bar chart version of this data, click the **Bar** icon (to the upper right of the table).

#### IMD Quality Trend

The **IMD Quality Trend** zone displays the number of Validation, Editing, and Estimation (VEE) exceptions and calculated numbers of measurements for each day in the selected time period. The x-axis displays individual dates during the selected period. The left y-axis shows the average and maximum number of measurements per initial measurement. The right y-axis shows the number of VEE exceptions. The graph shows the following:

- **VEE Exceptions**: The number of VEE exceptions processed on each day in the selected time period.
- **Average Measurements**: The average number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.
- **Maximum Measurements**: The maximum number of measurement values (for example, individual interval values) per initial measurement on each day in the selected time period. This is calculated by dividing the largest number of measurements in all of the initial measurements on the day in question by the number of initial measurements on the day in question. The number of measurements is based on the Start Date/Time, End Date/Time, and the interval size of each initial measurement.

You can filter the data displayed using the following controls:

- **Period View**: Specifies the range of data displayed. Options include Thirty Days, Seven Days, Three Days, or One Day. The default is "Seven Days".
- **Division**: Specifies the division of the initial measurements displayed in the graph. The default is "Show All".
- **Interval/Scalar**: Specifies whether the graph should display data for only interval initial measurements, scalar initial measurements, or both. The default is "Show All".
- **Business Object**: Specifies the initial measurement business object(s) related to the data displayed in the graph. The default is "Show All".
- **Excepti0on Type**: Specifies the exception type (Insufficient Input Data, Device Identifier Discrepancy, and so on) of the VEE exceptions displayed in the graph. The default is "Show All".


Copyright © 2026, Oracle and/or its affiliates.

- **Data Source**: Specifies the data source (if any) of the the data displayed in the graph. The default is "Show All".

Click on the labels on the legend below the graph to hide/show specific metrics.

This zone displays data from statistics snapshots based on the IMD Quality Snapshot - Hourly (D1-IMDQualityHourly) business object. The **Snapshot Date/Time** drop-down list (in the lower right corner of the zone) lists all currently available snapshots based on this business object. Filtering options are based on data available in the snapshot. For example, if all of the VEE exceptions processed for a specific snapshot are all based on interval initial measurements, the "scalar" option wouldn't appear in the **Interval/Scalar** drop-down list.

Other metrics, including File Count (the number of payload files processed) and Total Measurements, Average Age, and Maximum Age can be viewed from the statistics snapshot.

Click **Capture** (next to **Snapshot Date/Time** in the lower right corner of the zone) to navigate to the **Statistics Control** portal where you can view a list of snapshots for the statistics control, view details of a specific snapshot, and create a new snapshot if needed.

### IMD Quality Details

The **IMD Quality Detail** zone displays details from the most recent snapshot based on the IMD Quality Snapshot - Hourly (D1-IMDQualityHourly) business object. By default this zone displays a tabular view of the data from the snapshot, with each row in the table representing a unique combination of business object, interval/scalar flag, and status. This table includes the following:

- **Date Time**: The date and time on which the VEE exceptions captured in the row were processed
- **Business Object**: The business object of the initial measurements on which the VEE exceptions captured in the row are based
- **Interval /Scalar**: A flag designating if the VEE exceptions captured in the row are based on interval (D1IN) or scalar (D1SC) initial measurements
- **Exception Type**: The exception type of the VEE exceptions captured in the row
- **Description**: A description of the VEE exception type captured in the row
- **VEE Exceptions**: The number of VEE exceptions processed for the unique combination of business object, interval/scalar, and exception type captured in the row
- **Average Measurements**: The average number of measurements (Total Measurements divided by VEE Exceptions) per initial measurement on which the VEE exceptions captured in the row are based
- **Maximum Measurements**: The maximum number of measurements per initial measurement on which the VEE exceptions captured in the row are based
- **Total Measurements**: The total number of measurements values in the initial measurements on which the VEE exceptions captured in the row are based

To view a bar chart version of this data, click the **Bar** icon (to the upper right of the table).

### IMD Control Seeder Overview

The **IMD Control Seeder Trend** zone displays details from the most recent snapshot based on the IMD Control Seeder – Hourly (D1-IMDCtrlSeeder) business object. By default this zone displays a pie chart of the data from the snapshot, with each segment in the chart representing a unique combination of business object, interval/scalar flag, and message category and number.


Copyright © 2026, Oracle and/or its affiliates.

To view a table version of this data, click the **Grid** icon (to the upper right of the table). The tabular version of this data can also be seen in the **IMD Control Seeder Trend** zone.

### IMD Control Seeder Trend

The **IMD Control Seeder Trend** zone displays details from the most recent snapshot based on the IMD Control Seeder – Hourly (D1-IMDCtrlSeeder) business object. By default this zone displays a tabular view of the data from the snapshot, with each row in the table representing a unique combination of business object, interval/scalar flag, and message category and number. This table includes the following:

- **Date Time**: The date and time on which the initial measurements captured in the row were processed
- **Business Object**: The business object of the initial measurements captured in the row
- **Interval/Scalar**: A flag designating if the data captured in the row is interval (D1IN) or scalar (D1SC)
- **Message Category**: The message category for the exceptions associated with the IMD seeders captured in the row
- **Message Number**: The message number for the exceptions associated with the IMD seeders captured in the row
- **Message Text**: The message text for the exceptions associated with the IMD seeders captured in the row
- **IMD Records**: The number of initial measurements processed for the unique combination of business object, interval/scalar, and status captured in the row
- **Average Measurements**: The average number of measurements (Total Measurements divided by IMD Records) per initial measurement captured in the row
- **Maximum Measurements**: The maximum number of measurements per initial measurement captured in the row
- **Total Measurements**: The total number of measurements values in the initial measurements captured in the row

To view a bar chart version of this data, click the **Bar** icon (to the upper right of the table).

# Using the Service Order Operational Dashboard

This portal is used to display the Service Order Operational Dashboard.

You can access the portal from the **Main > Dashboards > Service Order Operational Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

The following zones may appear as part of the portal's **Main** tab page:

- **In-Flight Activities By Type**: shows the distribution of in-flight Activities (in progress or in error) by Activity Type.
- **In-Flight Activity Trend**: shows the number of activities that are in-flight on a given day.
- **In-Flight Activities with Issues**: presents a bar chart that indicates the number of activities with issues.
- **In-Flight Activity Status Distribution**: presents a pie chart that displays the distribution of statuses of in-flight activities.


Copyright © 2026, Oracle and/or its affiliates.

- **Efficiency Trend for In-Flight Activities**: presents a stacked bar chart that displays the distribution of in-flight activity processing efficiency relative to configurable tolerance.
- **Service Order To Do Summary**: this zone uses horizontal bars to highlight the relative number of open To Dos for a set of To Do Types related to Service Order Management.

Refer to About Dashboards for additional functional information. For information on configuration, refer to the related section for Configuring the Service Order Operational Dashboard in the Administrative User's Guide.

### Filtering by Division

Data displayed in the zones on this dashboard can be filtered by Division. To enable filtering by Division set the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration to "Yes". If this is set to "No" the zones display data for all divisions.

When filtering is enabled, the **Division** drop-down list allows the user to select the Division for which they wish to view data. Available divisions are based on the divisions assigned to the user in the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. The division marked "Use As Primary" is the default.

# Using the Service Order Trends Dashboard

This portal is used to display the Service Order Operational Dashboard.

You can access the portal from the **Main > Dashboards > Service Order Trends Dashboard**. Once you've selected this menu option, you're immediately brought to the initial tab of the dashboard.

The following zones may appear as part of the portal's **Main** tab page:

- **Orchestration Issues Trend**: contains a stacked bar chart showing issues by category and Activity Type.
- **Issue Distribution Over Time**: contains a pie chart showing the distribution of activity issues over a range of days.
- **Activity Success Trend**: shows successful activities over a range of days.
- **Activity Creation Trend**: shows activities created over a range of days.
- **Activity Completion Efficiency Over Time**: shows the efficiency of activity completion over a range of days.

Refer to About Dashboards for additional functional information. For information on configuration, refer to the related section for Configuring the Service Order Trends Dashboard in the Administration Guide.

### Filtering by Division

Data displayed in the zones on this dashboard can be filtered by Division. To enable filtering by Division set the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration to "Yes". If this is set to "No" the zones display data for all divisions.

When filtering is enabled, the **Division** drop-down list allows the user to select the Division for which they wish to view data. Available divisions are based on the divisions assigned to the user in the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal. The division marked "Use As Primary" is the default.


Copyright © 2026, Oracle and/or its affiliates.

# Totals and Trends


- Processing Statistics
- Consumption Statistics


## Processing Statistics


- About Processing Statistics
- About Head-End System Processing Statistics


### About Processing Statistics

Upload statistics (or upload stats) are statistics related to the uploading of initial measurement data and device events sent from a head-end system.

Upload statistics are defined as activities in Oracle Utilities Smart Grid Gateway.

There are three types of upload statistics activities:

**Payload Statistics**: Contains statistics related to a specific payload (file) containing one or more initial measurements or device events. Payload Statistics activities contain:

- Basic information about the payload (head-end system, file name, and status
- Middleware statistics including specifics about the file, the total number of initial measurements or device events processed, the number of initial measurement or device events errors, and total processing time
- Initial measurement statistics including the number of initial measurements processed
- Device event statistics including the number of device events processed

**Payload Error Notification**: Contains details concerning processing errors encountered in an individual payload (file) containing one or more initial measurements or device events. Payload Error Notification activities are related to Payload Statistics activities.

**Payload Summary**: Contains processing summary statistics for an individual payload (file) containing one or more initial measurements or device events. Payload Summary activities are related to Payload Statistics activities, and are used to update related payload statistics upon the completion of payload processing.

Upload statistics activities are created during processing of payload files as follows:

- When processing begins for a payload, a Payload Statistics activity is created to record the process.
- If an error occurs during processing, a Payload Error Notification activity is created.
- When payload processing is complete, a Payload Summary activity is created, which in turn, updates the Payload Statistics activity with details concerning the processing of the payload, including (the start and end time of the processing, the total processing time, the number of initial measurements or device events processed, and the number of initial measurement or device event errors (if any).



Copyright © 2026, Oracle and/or its affiliates.

## About Head-End System Processing Statistics

Head-end system processing statistics are summarized processing statistics for payloads received from a given head-end system.

As Oracle Utilities Smart Grid Gateway processes payloads containing initial measurements or device events, statistics for each payload are captured in Payload Statistics activities. Over time, payload statistics for each head-end system are summarized to allow administrators to view statistics summary for the head-end system.

Head-end system processing statistics are stored as aggregated measurements for aggregator measuring components. A separate aggregator measuring component must be set up for each head-end system for which processing statistics will be aggregated.

# Consumption Statistics

Please refer to the About Aggregation section for information on consumption statistics.

# Time Zone Support

- Daylight Savings Time Support
- Multiple Time Zone Support

# Daylight Savings Time Support

This section describes how the Oracle Utilities Service and Measurement Data Foundation and its related products support Daylight Saving Time (DST).

All date/times in the system are displayed adjusted for daylight savings time (when in effect). However, not all date/times in the system are stored in daylight savings time:

- Stored in Standard (not adjusted for DST): data that is highly sensitive to DST shifts are stored in standard time without any shifts, this means they have a consistent 24 hour day each day all year and avoid issues that arise from a skipped hour when transitioning to DST and a duplicate hour when transitioning out of DST. For example: initial measurement data, final measurements, device events, TOU map data.
- These date/times are adjusted for daylight savings time prior to being displayed to a user.
- Stored in Legal (adjusted for DST): data that is not as sensitive to DST time shifts are stored in legal time. For example, device configurations, service points, installation events, usage subscriptions, usage transactions.

**Adjusting Date/times Upon Receipt**

The two largest sources of imported data to the system can be sent in a mixture of date/time formats some in legal time some in standard time:

- Initial Measurement Data
- Device Events


Copyright © 2026, Oracle and/or its affiliates.

Since both sets of data are sent by head end systems and each head end system handles date/times differently, potentially even differing by device, the interface must be extremely flexible in how it handles date/times. In order to ensure data is accurately stored and processed the system must know how a given meter sends date/time information. This is defined on the device through the Incoming Data Shift field:

- Always in Standard Time: No special handling is required.
- Always in Legal Time: All date/times for the initial measurement data or device event will be converted from legal time into standard time.
- In circumstances where a date/time is received in legal time and it falls on the duplicate hour that occurs when DST ends special logic will be executed to identify whether that hour is the first (still in DST) or second (no longer in DST) of the duplicate hours. If for some reason it cannot be identified which hour it is the transaction will go into error.

### Daylight Savings Time Interval Conversion Examples

The following examples show how data received legal time will be shifted into standard time for an interval meter that is read on an hourly basis.

Daylight Savings not in Effect

| Date/Time Received in Legal | Date/Time Stored in Standard |
|---|---|
| 01/14/2010 12:00AM | 01/14/2010 12:00AM |
| 01/14/2010 1:00AM | 01/14/2010 1:00AM |
| 01/14/2010 2:00AM | 01/14/2010 2:00AM |
| 01/14/2010 3:00AM | 01/14/2010 3:00AM |
| 01/14/2010 4:00AM | 01/14/2010 4:00AM |
| 01/14/2010 5:00AM | 01/14/2010 5:00AM |

Daylight Savings Starts (Spring Forward) - No 2AM interval

| Date/Time Received in Legal | Date/Time Stored in Standard |
|---|---|
| 03/14/2010 12:00AM | 03/14/2010 12:00AM |
| 03/14/2010 1:00AM | 03/14/2010 1:00AM |
| **03/14/2010 3:00AM** | **03/14/2010 2:00AM** |
| 03/14/2010 4:00AM | 03/14/2010 3:00AM |
| 03/14/2010 5:00AM | 03/14/2010 4:00AM |
| 03/14/2010 6:00AM | 03/14/2010 5:00AM |

Daylight Savings in Effect

| Date/Time Received in Legal | Date/Time Stored in Standard |
|---|---|
| 06/14/2010 12:00AM | 06/13/2010 11:00PM |
| 06/14/2010 1:00AM | 06/14/2010 12:00AM |
| 06/14/2010 2:00AM | 06/14/2010 1:00AM |
| 06/14/2010 3:00AM | 06/14/2010 2:00AM |
| 06/14/2010 4:00AM | 06/14/2010 3:00AM |
| 06/14/2010 5:00AM | 06/14/2010 4:00AM |


Copyright © 2026, Oracle and/or its affiliates.

Daylight Savings Ends (Fallback) - Duplicate 1AM

| Date/Time Received in Legal | Date/Time Stored in Standard |
|---|---|
| 11/7/2010 12:00AM | 11/6/2010 11:00PM |
| **11/7/2010 1:00AM** | **11/7/2010 12:00AM** |
| **11/7/2010 1:00AM** | **11/7/2010 100AM** |
| 11/7/2010 2:00AM | 11/7/2010 2:00AM |
| 11/7/2010 3:00AM | 11/7/2010 3:00AM |
| 11/7/2010 4:00AM | 11/7/2010 4:00AM |

### Daylight Savings Impact on Periodic Estimation

In periodic estimation both the interval and scalar versions can be impacted by daylight savings time:

- Interval periodic estimation using the "Predictable Cut-off" method.
- All scalar periodic estimation

Refer to About Periodic Estimation for more information about the impact of Daylight Savings Time on periodic estimation.

# Multiple Time Zone Support

Some utilities have operations in more than a single time zone. When operating in multiple time zones, Oracle Utilities Meter Data Management and Smart Grid Gateway provide the following high level functionality:

- The ability to receive data in any time zone and convert it to the common time zone used by Oracle Utilities Meter Data Management (as defined by the base time zone)
- The ability to display functional date/times in the time zone of a given device, service point, usage subscription
- The ability to define what time zone a measuring component, device configuration, service point, or usage subscription resides in
- The ability to aggregate data across time zones into a single time zone or without respect to a time zone (see Aggregations on page E-17 for more information).
- The ability to export data to an external system with the appropriate time zone information to enable the external system to consume data from different service points in different time zones

By default, the system assumes only a single time zone is supported. To enable support for multiple time zones it requires that adding the appropriate entry to the general system configuration feature configuration.

Refer to Configuring Feature Configurations for additional on the feature configuration options available.

### Date/Time Storage and Display

All date/times in the system are stored in the base time zone. Date/times are stored in either standard time or legal time and this determination is made based on the type of data (e.g. measurements are in standard but installation event date/times are in legal).

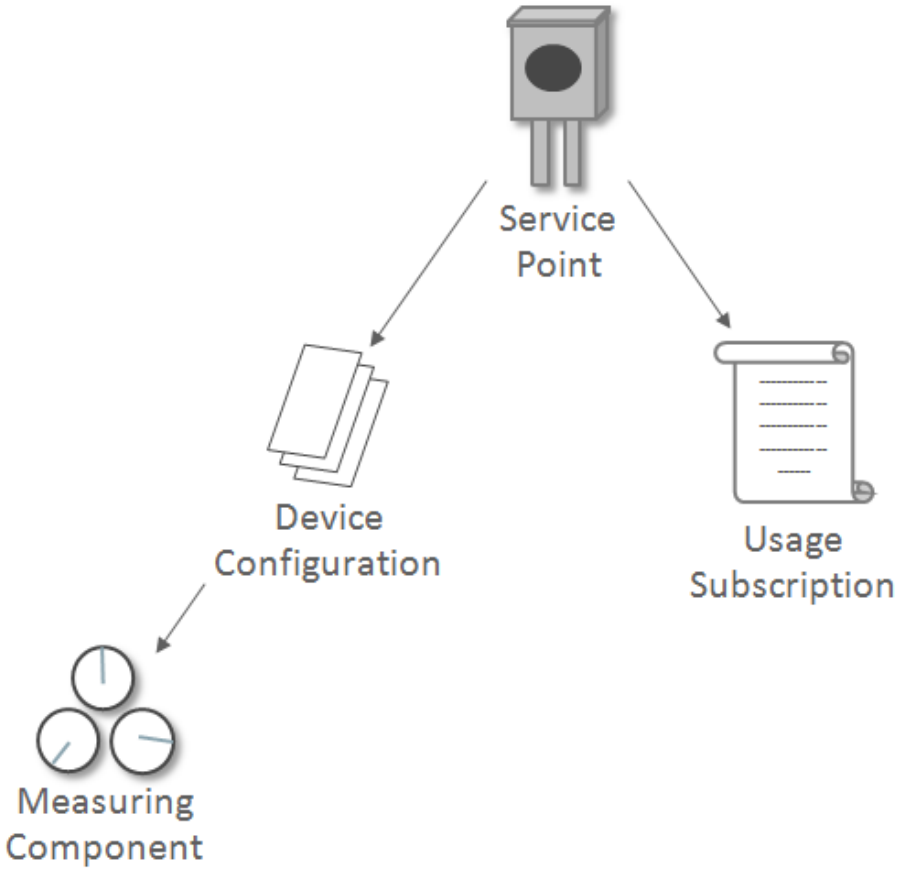

Copyright © 2026, Oracle and/or its affiliates.

All date/times are displayed in legal time. Those date/times that are key functional date/times related to a device, service point, or usage subscription will be displayed in the legal time of the time zone the device resides in.

Refer to the Glossary of Terms for a more detailed definition of Standard and Legal time.

**Time Zone Hierarchy and Maintenance**

A time zone can be defined at many points within the device to usage subscription hierarchy. This enables the time zone to be accurately portrayed based on the installation status of that device (e.g. not installed, installed but not billed, billed). At each stage of the hierarchy the time zones must match.




Once the hierarchy has been established no one portion of that hierarchy can be changed to a time zone that differs from the other levels of the hierarchy. In order to change the time zone for an established device to usage subscription hierarchy the change must be done to the service point. Once the service point time zone has been changed the system will cascade that change to all other levels of the hierarchy.

If a device has been installed at a service point in order for it to be moved to a different service point in a different time zone. It will require that a new device configuration be created for time zone of the new service point since a device configuration cannot be associated to two different service points each with a different time zone from the device configuration.

Note: If the time zone for a device that has existing measurements has been changed there will be no automatic adjustments made to the underlying measurement data. If the measurement data must be changed it will require manual intervention.


Copyright © 2026, Oracle and/or its affiliates.

Note: Although measuring component has time zone it is not populated on physical devices that are installed at service points. This time zone is used for standalone measuring components.

### Impacts on Aggregations

In aggregation there are two use cases for how data should be aggregated for service points that exist in differing time zones:

- **Absolute Time**: If a time zone is provided on the measuring component or measuring component type, then the aggregation horizon date/times will be used across service points of differing time zones. For example the 4/1/2010 12AM interval in US/Pacific would be combined with the 4/1/2010 3AM interval in US/Eastern. Since all date/times are stored in the base time zone there is no need to do time zone conversions.
- **Local Time**: If no time zone is specified, consumption is aggregated by converting the aggregation horizon to the appropriate time zone for each service point. For example the 4/1/2010 12AM interval in US/Pacific would be combined with the 4/1/2010 12AM interval in US/Eastern.

Aggregation Measuring Components in general do not have any time zone related dependencies with other master configuration objects. However, when an aggregator is listed as a direct channel measuring component on a usage subscription it share the same time zone as the Usage Subscription.

### Impacts on TOU Map Generation

The date and times defined on a Time of Use Mapping Templates (TOU) do not have a time zone associated to them and will be interpreted based on the time zone of the TOU map type of the specific TOU Map being processed. This allows for a single definition of a TOU schedule that can be applied across many different time zones.

During the generation of the TOU map data for a given TOU map template the date/times will be converted from the time zone of the TOU map to the time zone defined in the base time zone defined on the installation options - framework. For example, if 1/1/2010 5:00PM was considered to be "On Peak" and the system time zone was UTC -08:00 and the TOU map time zone was UTC -05:00 then the TOU map data entry would be written as 1/1/2010 2:00PM.

### Impacts on Factors

Factor effective dates do not have a time zone associated to them and are assumed to be in the local time zone of the data being validated. For example a factor effective 4/1/2010 12:00AM would be considered to be effective 4/1/2010 12:00AM UTC -08:00 and 4/1/2010 12:00AM UTC -05:00 based on the time zones of the respective meters. This allows a single factor to apply to many time zones and behave uniformly.

### Impacts on Holidays

Holiday effective dates do not have a time zone associated to them and are assumed to be in the local time zone of the data being validated. For example a holiday on 12/25/2010 would be considered to be effective beginning 12/25/2010 12:00AM UTC -08:00 and 12/25/2010 12:00AM UTC -05:00 based on the time zones of the respective meters.

### Impacts on Periodic Estimation

Refer to About Periodic Estimation for more information about the impact of Daylight Savings Time on periodic estimation.


Copyright © 2026, Oracle and/or its affiliates.

### Impacts on Initial Measurement Creation Online

When creating initial measurements online from the 360 Degree View portals, initial measurements are created as appropriate for measuring components with time zones other than the base time zone.

For example, if you copy measurements from a measuring component in one time zone to a measuring component in a different time zone, the measurements are being moved in absolute time (i.e. the 3:00 PM in UTC -08:00 would be changed 6:00PM in UTC -05:00).

### Impacts on Items

The consumption for "badged" and "unbadged" items is driven by configuration on the device type service quantity and the multi-item section of the service point:

- **Device Type Service Quantity**: the effective date of the service quantity does not have a time zone and is assumed to be in the local time zone of the service point it is associated to (whether through a install event for "badged" items or by being directly configured on the service point). For example an effective date time of 1/1/2010 5:00PM would be considered to be effective beginning 1/1/2010 5:00PM UTC -08:00 and 1/1/2010 5:00PM UTC -05:00 based on the time zones of the respective meters
- **Service Point Multi-Item**: the start date time and end date time of the multi-item list are stored in legal time of the base time zone and displayed in the legal time of the service point's time zone.

# Service Issue Monitors


- About Service Issue Monitors
- About Service Investigative Orders


## About Service Issue Monitors

Service Issue Monitors are service tasks that analyze service points to determine if service is needed. If service is determined to be needed, the Service Issue Monitor creates a Service Investigative Order.

Device events, VEE exceptions, failed smart meter commands, and other events can trigger the creation of a service issue monitor (the type of service issue monitor created is based on the Service Issue Monitor Type specified on the device event type, exception type, or activity type). Once created, service issue monitors analyze the service point where the device associated with the device event, VEE exception, or failed command, based on evaluation criteria specified on the service issue monitor's type. If the criteria are met (in other words, if a specified number of command failures, device events, or VEE exceptions are found for the service point), the service issue monitor creates a service investigative order.

### Service Issue Monitor Creation

This section provides an overview of the process that takes place when service issue monitors are created.



Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

1. One (or more) of the following takes place:
   - A device event or reader remark is received, whose type specifies a Service Issues Monitor Type
   - A VEE exception whose type specifies a Service Issues Monitor Type is generated during VEE processing
   - A usage transaction exception generated during usage calculation
   - A smart meter command fails whose type specifies a Service Issues Monitor Type
   - A business flag is received whose type specifies a Service Issue Monitor Type
2. The object received in item (1) creates a service issue monitor based on the referenced Service Issue Monitor Type
3. The service issue monitors analyzes the service point based on the Discard Rules and Evaluation Criteria defined on the service issue monitor's type to determine if a service investigative order should be created or if the service issue monitor should be discarded.

   The "Discard Rules" on the service issue monitor's type are evaluated to determine if the new service issue monitor should be discarded (new service issue monitors are always discarded when created if an existing service investigative order initiated by a service issue monitor of the same type exists for the service point.)

   If the service issue monitor type was configured to require approval, the service issue monitor generates a To Do to notify the appropriate individuals that approval is necessary.

   If the service issue monitor is approved or if no approval was necessary the service investigative order will be continue.
4. The service issue monitor then executes the evaluation criteria defined on the service issue monitor's type. If those criteria are met the Service Investigative Order activity is created. If the criteria are not met the service issue monitor is discarded.
5. If the service issue monitor has completion criteria defined the service issue monitor will wait for the service investigative order to complete. Once complete the the service issue monitor will execute the completion criteria to evaluate if the service investigative order achieved the desired results.

   If either there were no completion criteria or they were satisfied the service issue monitor will complete.

   If the completion criteria were not satisfied the service issue monitor enter an inconclusive investigation state from which manual intervention will be required.

# About Service Investigative Orders

Service investigative orders are activities created by a service issue monitor when a specified set of events have occurred at a service point. The type of activity created by the service issue monitor is defined on the service issue monitor's type.

Service issue monitors are often configured to create field activities that are in turn sent to an external field work system.


Copyright © 2026, Oracle and/or its affiliates.

# Service Order Management


- About Service Order Activities
- About Service Order Field Activities
- About Appointments


## About Service Order Activities

Service order activities represent specific service order requests either received from an external system or created from within Oracle Utilities Smart Grid Gateway.

Service order activities are created when specific types of requests for service are received in the system. These activities orchestrate the steps necessary to fulfill the request, based on the current state of the service point related to the request. For example, the system creates an "enable service" activity when a request is made to enable service at a particular service point. This activity examines the current state of the service point, and creates one or more additional activities if appropriate. For example:

- If there is no meter currently installed at the service point, the service order activity would create an "install meter" service order field activity and send it to a field work system.
- If there is a meter installed, but the meter is not currently registered with the head end system, the service order activity might create a "commission meter" smart meter command.
- If the meter is active and commissioned, but not currently connected, the service order activity might create a "remote connect" smart meter command.
- Etc.

Service order requests are typically received from the customer information system such as Oracle Utilities Customer Care and Billing, but can also be created from within Oracle Utilities Meter Data Management or Oracle Utilities Smart Grid Gateway.

The base package provides the following types of service order orchestration activities:

- Enable Service: Used to enable service at a service point.
- Disable Service Used to disable service at a service point.
- Cut for Non-Payment: Used to cut off service at a service point due to non-payment of past due amounts.
- Reconnect Service for Payment: Used to restore service at a service point after receipt of past due payment.
- Exchange Meter: Used to orchestrate the exchange of meters at service point, such as in the event that a customer upgrades their meter.
- Back to Back Service: Used to orchestrate a change of service when the customer at a service point changes (such as when owners/tenants change).

Service order orchestration activities are defined by the following:

- **Activity Type**: The service order's activity type.
- **Service Date/Time**: The date and time the service order request was received.



Copyright © 2026, Oracle and/or its affiliates.

- **Service Point**: The service point associated with the service order request.
- **Service Agreements**: A list of one or more service agreements related to the service order request (not used with all service order orchestration activities).
- **Request Information**: Details of the service order request, including requester and external system information.
- **Collection Data Details**: Details regarding past due balances (used with Cut for Non-Payment activities only).
- **Contact Details (or Customer Information)**: Contact details for the customer associated with the service order request.

Note: Some service order activities use additional data.

In the event that an active service order activity needs to be updated and/or cancelled, the system also supports the following maintenance service order activities:

- Cancel Orchestration: Used to cancel a currently active service order activity.
- Update Orchestration: Used to update a currently active service order activity.

Service order maintenance activities are defined by the following:

**Activity Type**: The service order's activity type.

**Orchestration Activity**: The original orchestration activity related to the maintenance activity.

**Related Activity**: A specific activity (such as a service order field activity or smart meter command) related to the maintenance activity.

**Cancel Reason**: The reason for cancelling the original activity (for cancel activities only).

**Comments/Instructions**: Specific comments and/or instructions applicable to the maintenance activity (for update activities only).

**Start Date/Time**: The date and time the maintenance activity was initiated (for update activities only).

**Request Information**: Details of the service order request, including requester and external system information.

**Activity Hierarchy**: A tree that displays the activity and related activities (including service order field activities and smart meter commands), inbound and outbound communications, and completion events.

# About Service Order Field Activities

Service order field activities are specific types of tasks performed by field personnel. Examples include installing meters, repairing/replacing meters, etc.

Service order field activities are created when completing specific types of requests for service requires field work. Once created, service order field activities are sent to a field work system, where they are scheduled and assigned to a field work crew. When the task is completed by the crew, the field work system sends an response to the system and the service order field activity is updated as appropriate.

When communicating with field work systems, service order field activities create outbound communications to send the required information to the field work system. When the field work system sends its response, the system creates an inbound communication. See Communications for more information about communications. Service order field activities that


Copyright © 2026, Oracle and/or its affiliates.

have created outbound communications can also be manually completed or canceled if necessary.

Service order field activities are created when a service order orchestration activity examines the current state of the service point, and determines that field work is required to complete the service request. For example, if an "enable service" activity determines that there is no meter currently installed at the service point, the service order activity would create an "install meter" service order field activity and send it to a field work system.

Service order field activities are defined by the following:

- **Activity Type**: The service order field activity type.
- **Status**: The current status of the service order field activity.
- **Service Date/Time**: The date and time the service order field activity was created.
- **Service Point**: The service point associated with the service order field activity.
- **Field Task Type**: The field task type for the service order field activity. The list of available field task types can be filtered based on the **Valid Field Task Types** defined for the service point's service point type via the **Restrict Field Task Type by Service Point Type** option on the Service Order Management Master Configuration.
- **Recipient**: The service provider to which the service order field activity is sent for scheduling and assignment.
- **Device**: The device related to the service order field activity (if applicable).
- **Request Information**: Details of the service order request, including requester and external system information.
- **Contact Details (or Customer Information)**: Contact details for the customer associated with the service order request.
- **Address Information**: The address of the service point associated with the service order field activity.

# About Appointments

The Appointment Booking portal is used to request and track appointments related to service order field activities that have been scheduled in an external field work scheduling system.

When service order field activities send outbound communications to the field work system, appointments are created in field work system so that the work can be assigned to a crew. Requests for appointments are most often requested automatically when the service order field activity is created, but if needed, users can request appointments manually.

Users work with the Appointment Query portal to search for activities and related appointments.

Users work with the Appointment Booking to request and track appointments for service order field activities. This portal includes the following zones:

- **Service Point Activities**: Displays a list of activities and related appointments for the current service point in date descending order. This zone also includes a filter area which allows the user to define the inclusion or exclusion of activities according to certain criteria
- **Activity Summary**: Displays details of a selected activity in the Service Point Activities zone (indicated by a star).


Copyright © 2026, Oracle and/or its affiliates.

- **Appointment Booking Results**: Displays a list of available appointment slots as returned by the scheduler. This zone allows the user to book an appointment by selecting a slot and subsequently updating the orchestration or field activity.

# Data Access Restrictions


- Using Data Access Restrictions


## Using Data Access Restrictions

The focus for multi-jurisdiction support in Oracle Utilities Meter Data Management is on restricting access to the user interface. This occurs through defining different Access Groups by operating company which then marks the Service Point with the access capabilities. The Service Point has a field named **Access Group** which allows you to define specific access to the Service Point.

If Access Group is set on the Service Point, then the user interface search screens within Oracle Utilities Meter Data Management will only allow users that are assigned to the proper Access Role to be able to access that Service Point.

For additional information see Defining Access Groups.

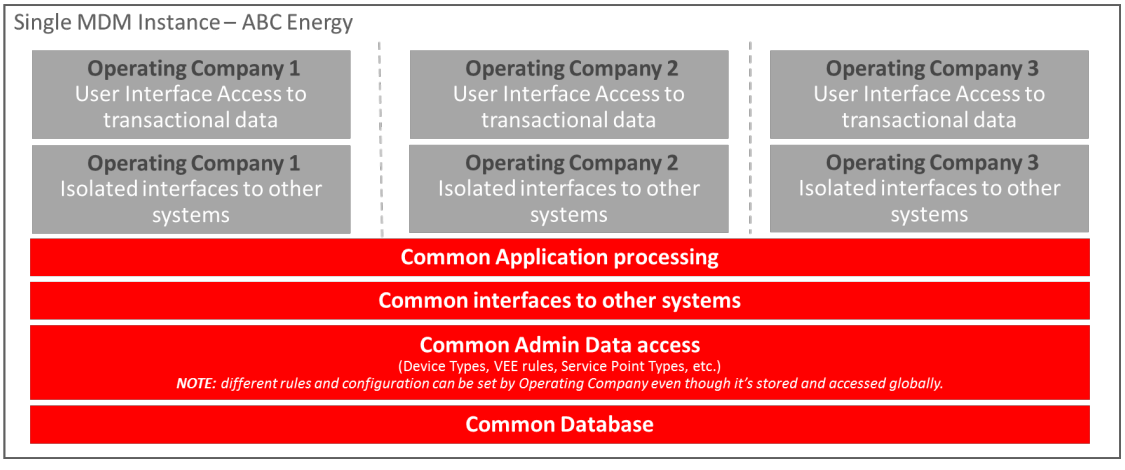
### Defaulting Access Groups on Service Points

If there is a desire to set a default Access Group by jurisdiction, then this can be accomplished on configuration for Markets. When defining a Market there is an option to add a **Default Access Group**. This value will be used to mark any newly created Service Point with the same Access Group value defined on the related Market.

### Example Scenarios

Below are some examples of how Data Access Restrictions can be implemented:

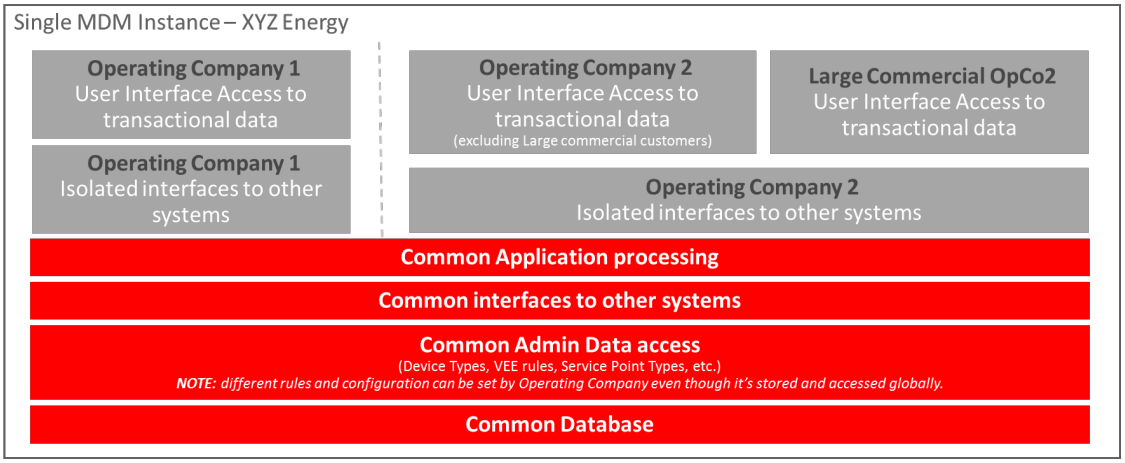
**Example 1: Split Access by Market** - an example utility "ABC Energy" that has 3 operating companies.




**Example 2: Isolating Important Customers** - an example utility "XYZ Energy" that has 2 operating companies but would also like to isolate access to Large commercial customers for one of the companies.



Copyright © 2026, Oracle and/or its affiliates.




# To Do Processing

There are numerous events that trigger the creation of email-like messages describing work that requires human attention. For example, if a bill segment has an error, a message is created to highlight such. We refer to each message as a **To Do Entry**. In this section, we describe how to view, manage and work To Do entries.

> **NOT_SUPPORTED**
>
> We recommend familiarizing yourself with the topics described in The Big Picture of To Do Lists to fully appreciate how To Do entries are managed by the system.

- Understanding To Do Processing
- Managing To Do Entries

## Understanding To Do Processing

The topics in this section provide background information on To Do processing functionality.

- To Do Role
- To Do Priority
- Work Distribution
- Working On A To Do
- To Do Characteristics
- Related To Do Entries
- Complete A To Do
- Log Entry Events

### To Do Role

Each To Do entry is assigned a specific To Do Role. The role defines the users who may work on the entry. The permissible roles that may be assigned to a To Do entry are defined on the


Copyright © 2026, Oracle and/or its affiliates.

entry's To Do type. One or more users within a role may be designated as a supervisor, who is allowed to assign To Do entries to other people. Supervisors are determined through application security access.

Refer to To Do Entries Reference A Role to learn more about role assignment.

After an entry is created, its role may be changed to any role defined as valid for the entry's To Do type. Manual changes of a To Do entry's role are recorded on the To Do entry log.

## To Do Priority

A To Do entry is associated with a priority level representing its processing order compared to other entries. Refer to The Priority Of A To Do Entry for more information on how an entry's priority is initially assigned and how it may be automatically updated by the system.

A user may manually override a To Do entry's priority at any time. Once overridden, automatic recalculation of the priority is no longer performed so as not to override the value explicitly set by the user. Manual overrides of a To Do entry's priority are recorded on the To Do entry log.

> **Note**
>
> Only users that are currently **Assigned** to a To Do entry or ones that are valid for the role of an **Open** To Do entry may manually change the priority.

## Work Distribution

This topic goes overs the various possible actions a user can take to adjust what role or what user is associated with a To Do entry while it is in a non-complete state.

**Adjusting the Role**

A To Do entry must always reference a role. When a To Do entry are created in the **Open** state and is associated with a role. The following points highlight ways to adjust the role associated with a To Do entry.

- A user viewing a record on the To Do Entry page may use the Forward or Send Back action to change the role.
- A supervisor using the To Do Search page may assign a user not linked to the current role. In this case, the system will update the role to one valid for the assignee.

**Assigning a To Do to a User**

There are several ways to **Assign** a To Do entry to a user:

- The To Do Management portal provides many filters and viewing options to review To Do entries for a give To Do Type. In addition, it displays a rich amount of detail for the resulting To Do entries, helping users to make decisions. Users may use this page to assign work to themselves and supervisors may use this to assign work to others.
- A user viewing an entry on the To Do Entry page may use the Forward to a User or the Send Back action to assign the entry to a different user.
- **On demand assignment**. The system can be configured to dynamically assign the highest priority work to a user on demand from the Assign me a To Do button on the Current To Do Zone. One of the main benefits of the assignment on demand approach is that work is allocated real-time based on most current information such as skills, priority, user


Copyright © 2026, Oracle and/or its affiliates.

availability, etc. This also means that work remains unassigned until a user is actually going to work on it making it available to any qualified user.

> **Note**
>
> Refer to On Demand Assignment Configuration for information on configuring the system to support this feature.

- Several legacy pages include assignment capability
  - A supervisor can use the Supervisor Functions pages to review, assign and reassign To Dos to users.
  - A user may use the To Do list pages to review and assign To Dos to themselves.
  - All of the above assignment functions may also be performed using the To Do Search page.

**Reopening / Unassigning a To Do**

It is possible to transition a To Do entry from **Assigned** back to **Open**.

- A user viewing a record assigned to them on the To Do Entry page may use the Forward to a role action to move the entry back to **Open**.
- The **Unassign** button on the To Do Management portal or the **Reopen** button on the To Do Search page causes the To Do entry to go back to **Open**.

# Working On A To Do

The To Do entry you are actively working on is your **In Progress** To Do entry. This To Do entry is visible in the Current To Do zone. Refer to that zone's help topic for all the features enabled on that zone.

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

If you don't currently have an **In Progress** To Do entry, there are several ways to choose a To Do and indicate that you want to start working on it. Doing so proactively populates the To Do into the Current To Do zone and navigates you to the related object (or if the To Do type is configured to launch a script, the script is launched).

If your organization's practice is to assign work to users up front, then a typical approach is that you would start by reviewing a list of entries assigned to you using one of the base pages and indicate that you want to work on an entry in the list. Or your organization may allow you to review To Do entries that are open and in a role that you are in. In some pages, there is a way for you to enable a list of To Do Entries that you can work through without needing to return to the query page to get the next entry.

The following points highlight the available base pages and their supported functionality.

- To Do Management allows you to use filters to view a list of entries assigned to you or that are open. Click the Work List icon in the Action column to populate the Work List zone in


Copyright © 2026, Oracle and/or its affiliates.

the sidebar with the results of the search. Click the **Work** button to choose the To Do entry you wish to start with. This drills you into the related object and populates the Current To Do zone with this To Do. It marks the To Do **In Progress**. (If the To Do entry was not already assigned to you, it will first assign it to you). Use the entries in the Work List zone to subsequently work on other To Do entries.

- If you are viewing a record on To Do Entry that you are allowed to work on, click the **Work** button to indicate that you want to start working on the record. This drills you into the related object and populates the Current To Do zone with this To Do. It marks the To Do **In Progress**. (If the To Do entry was not already assigned to you, it will first assign it to you).
- To Do List allows you to start working on a To Do by clicking the entry's message description link. It marks the To Do **In Progress**. (If the To Do entry was not already assigned to you, it will first assign it to you). Any other To Do entries that were in that results list on that page become an internal work list and the **Previous** button and **Next** button are enabled in the Current To Do zone to work through that list. Note that clicking the next or previous button does not automatically assign the entry to you. Use the **Start Progress** button on the Current To Do zone to change the status. If you don't start progress, it means the To Do in this zone is not your current In Progress to do entry. It remains in this zone until you perform another action to populate a different To Do in this zone. If you open another browser window, the Current To Do zone will not have this same To Do entry populated.
- To Do Search allows you to drill into the details of a To Do by clicking the entry's message description link, but it does not automatically assign it to you or update it to **In Progress**. You can click **Start Progress** on the Current To Do zone once it's populated. Any other To Do entries that were in that results list on that page become an internal work list and the **Previous** button and **Next** button are enabled in the Current To Do zone to work through that list. As with the first To Do selected, the subsequent To Do entries will not be assigned to you or progress started automatically.

> **Note**
>
> The To Do Search works differently than other pages in that you can drill in on a To Do entry and push the details into the Current To Do zone even if the To Do entry is assigned to someone else or if you are not allowed to work on that To Do. If you have an **In Progress** To Do entry and use the To Do Search to drill in on a different To Do Entry that doesn't get assigned to you, that To Do is now the one in the Current To Do zone and remains there until you do something else to refresh the data in that zone. If you open another browser window, the Current To Do zone will not have this same To Do entry populated.

Refer to Current To Do for more information about actions available to update To Do entries that you are working on.

Additionally, you may have the option to click on the **Assign me a To Do** button on the Current To Do zone to request your next task. The returned To Do entry automatically becomes the your **In Progress** To Do. This does not enable any work list functionality.

> **Note**
>
> If you do already have an **In Progress** To Do and you perform one of the above actions that results in a different To Do to become **In Progress**, the To Do entry that was already **In Progress** is updated to **On Hold**.


Copyright © 2026, Oracle and/or its affiliates.

> **Important**
>
> If your implementation is not taking advantage of the substatus as described in The Lifecycle Of A To Do Entry, then the system doesn't have a persistent way of marking which To Do entry is the one you are working on. You may still perform any of the steps described above to make the To Do entry your 'current' To Do. If any of the above steps indicate that the To Do will be assigned to you, that action still occurs, but the To Do entry is not moved to 'in progress'. The To Do entry is only considered the 'current To Do' for your current session. If a different session is opened, you must perform the explicit action again to make the To Do the 'current To Do' for that session.

## To Do Characteristics

Characteristics are used to associated related information to a To Do entry. Refer to Linking Additional Information to a To Do Entry for more information.

Several pages in the system allow you to view a To Do entry's characteristic values and / or search for To Do entries by their characteristics.

- The To Do Management portal and the To Do Details tab on the To Do Dashboard portal include up to five filters to search for To Do entries by a characteristic value. In addition, the search results display all the current characteristics as name / value pairs.
- The To Do Entry includes a zone to display any existing characteristics. In addition, if a user is allowed to modify a To Do Entry, the edit dialogue allows a user to add or change characteristics. Note that for manual To Do types, the valid characteristic types are driven by the To Do type. For automated To Do types, if valid To Do characteristic types are configured on the To Do type, only those are available. Otherwise, any valid To Do characteristic type may be linked.
- The To Do Search includes a single characteristic type / value filter.

## Related To Do Entries

A user working on a To Do may want to know of other To Dos that may be related to the same problem. If they can be worked by the same user they may get resolved faster. It seems almost impossible to programmatically determine whether two To Dos really stem from the same problem. The user should therefore be presented with a list of potentially related To Dos from which they can select the truly relevant ones.

A To Do entry is considered potentially related to a given To Do if they reference a common entity via a characteristic. For example, if a given To Do references a certain account then any To Do referencing that same account is considered related to it.

> **Note**
>
> The common entity is any characteristic value linked to the To Do entry where the characteristic type is a **Foreign Key** and the main table of the foreign key is classified as a **Master** or **Transaction** table.

There are various places where the user is informed as to whether or not the current To Do Entry is related to other To Do Entries.


Copyright © 2026, Oracle and/or its affiliates.

- To Do entry maintenance displays in insight indicating the number of related To Do Entries for the current To Do. In addition, a Related To Do Entries zone shows information about each related To Do.
- The Current To Do zone includes related To Do information and actions.
- The To Do Management query includes a count of related To Do entries.
- The To Do Search page supports a **Related To Do filter** that lists all related To Do entries for a given entry.

## Complete A To Do

After you finish working on an entry, you can mark it as **Complete**. At this point, you may want to review potentially related To Do entries and complete those that have been addressed by resolving the problem associated with the current entry.

When you complete a To Do, a log entry is created that indicates when the To Do was completed and by whom.

There are multiple ways in which you can complete a To Do.

- The To Do Management page provides many filters in addition to To Do Type. One or more entries may be completed in this page.
- Use the To Do Entry page to complete a single To Do Entry. The user may also choose to add comments when completing a To Do on this page.
- You may use the Current To Do zone to complete the current entry you are working on and/or all related entries assigned to you.
- The To Do List page allows you to complete one or more selected entries of a specific To Do Type.
- The To Do Search page allows you to search for To Do entries using various search criteria and complete selected ones.

> **Note**
>
> It is possible for the system to automatically mark an entry as **Complete**. Refer to The Lifecycle Of A To Do Entry for more information.

## Log Entry Events

Any user may add a **User Details** log entry to any To Do at any time (even after it's completed). Refer to To Do Entry page for more information.

The To Do Management portal also allows users to add a **User Details** log entry to one or more selected entries.

In addition, there are several log entries created automatically by the system. The log includes a status column, which captures the status of the To Do at the time of log. If the To Do entry is **Assigned** and its substatus **In Progress** or **On Hold**, the log captures the substatus value, otherwise it captures the status value.

The following table contains the list of all possible log events.


Copyright © 2026, Oracle and/or its affiliates.

| Event | Log Type | Log Status | Content of Details | Content of Assigned To (on To Do entry and To Do Log) | Log User ID | Contents of Sent By (on To Do entry) |
|---|---|---|---|---|---|---|
| A To Do is created (by the system or by a user) | **Created** | **Open** | | | ID of user who created the To Do | |
| A To Do is created (by the system or by a user) in Assigned status | **Created** | **Assigned** | | User ID of user who was assigned this To Do | ID of user who created the To Do | Creating user's ID |
| A To Do is completed (by the system or by a user) | **Completed** | **Complete** | | | ID of user who completed the To Do | |
| A user takes an open To Do | **Assigned** | **Assigned** | Selected by user | Assigned To User ID | Assigned To User ID | ID of user who assigned the To Do |
| A supervisor assigns a To Do to a user (also happens if they reassign) | **Assigned** | **Assigned** | Assigned to user | Assigned To User ID | ID of user who assigned the To Do | ID of user who assigned the To Do |
| A user forwards an entry to a user | **Forwarded** | **Assigned** | *Log details entered by user* | Forwarded To User ID | ID of user who forwarded the To Do | ID of user who forwarded the To Do |
| A user forwards an entry to a role | **Forwarded** | **Open** | Forwarded to role *Role Description (Role ID)* + *Log details entered by user* | | ID of user who forwarded the To Do | ID of user who forwarded the To Do |
| A user sends back a To Do | **Sent back** | **Assigned** | *Log details entered by user* | Sent Back To User ID | ID of user who sent back the To Do | ID of user who sent back the To Do |
| The status is updated (aside from the cases that have an explicit log type) | **Status Updated** | Current status of To Do | | User currently Assigned, if applicable | ID of the user the made the status update | |
| A user adds a log entry | **User Details** | Current status of To Do | *Log details entered by user* | | ID of user who added the entry | |


Copyright © 2026, Oracle and/or its affiliates.

| Event | Log Type | Log Status | Content of Details | Content of Assigned To (on To Do entry and To Do Log) | Log User ID | Contents of Sent By (on To Do entry) |
|---|---|---|---|---|---|---|
| A user manually overrides priority | **Override Priority** | Current status of To Do | Priority changed from *old value* to *new value* | Assigned To User ID | ID of user who made the change | |
| Priority is updated by a calculate priority algorithm | **Calculated Priority** | Current status of To Do | Priority changed from *old value* to *new value* | | ID of user that added or updated the To Do or the user associated with the Recalculate Priority batch job. | |

# Managing To Do Entries

This section describes the various pages available to view and work on To Do entries.

- To Do Management
- To Do Dashboard
- To Do Entry
- Legacy UI Features

## To Do Management

The To Do Management portal provides a multi-query zone that includes a search for To Do entries using To Do Type and other To Do information. Users may search for entries using multiple criteria and perform common actions, such as assignment and completion, on selected entries.

It is intended to be used by both supervisors and non-supervisor To Do users. Supervisors are identified by their security access to the portal.

Use **Menu**, and then **To Do**, and then **To Do Management** to navigate to the portal.

### To Do Information Search

This zone allows you to search for non-closed To Do entries using various attributes of the entries, and to perform actions on selected ones.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Results are restricted by role.** For supervisors, this search returns To Do entries that meet the search criteria and are in a role of which the supervisor is a member. For non supervisors, the search returns open To Do entries that meet the search criteria and are in a role of which the user is a member or entries currently assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

The following sections describe the features of this zone.

### Search Filters

The following table describes each of the different filters supported by this zone.

| Search Filter | Description |
|---|---|
| To Do Type | Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection. If the To Do Role filter is populated, the list is further restricted to valid To Do Types for the To Do Role. |
| Status | Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries. |
| Days Old | On the **To Do Management** portal, this option is represented by a pair of filter fields. Use these filters to restrict the search to To Do entries whose age falls within a given range in days. For example, entering values of 2 - 5 will restrict the search to entries that are greater than or equal to two days old and less than or equal to five days old.<br>On the **To Do Dashboard** portal, this option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days. |
| To Do Role | Use this filter to restrict the search to To Do entries of a specific role. Only To Do Roles where the user is a member will be available for selection. If the To Do Type filter is populated, the list is further restricted to valid To Do Roles for the To Do Type. |
| Assign To User | Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access. |
| Priority | Use this filter to restrict the search to To Do entries with a specific priority. |
| Key Value | Use this filter to restrict the search to To Do entries with a sort key value containing the entered value. |
| Message | Use this filter to restrict the search to To Do entries with a specific message. This filter appears after the To Do Type filter has been populated and shows a list of the message texts from all non closed To Do entries of the selected To Do Type. |


Copyright © 2026, Oracle and/or its affiliates.

| Search Filter | Description |
|---|---|
| Characteristic | Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. If the To Do Type filter is populated and the To Do Type is 'manual', the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do type, only those are shown. If an automated To Do Type doesn't define valid characteristics or if no To Do Type filter is populated, all valid characteristic types for the To Do Entry entity are shown.<br>Note that if you have populated characteristic filters and then you change the To Do Type filter, the system retains the characteristic filter as long as the characteristics are applicable to the newly selected To Do Type. |

The system shows the total number of To Do entries displayed and the total number that satisfy your search criteria immediately below the set of results.

> **Note**
>
> For performance reasons, because the filter criteria may be broad, the search zone is configured to only get the first 5000 records that match the criteria and will then sort that number to then display the first 400 sorted by priority and age. The To Do Dashboard portal may be used to get exact counts of records with certain criteria. Use additional criteria to narrow down the search as needed.

### Search Results

The zone will display all To Do entries that meet the search criteria, up to the limit defined. The following columns are displayed initially:

- The Action column includes the button Work. Click this button to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

> **Important**
>
> **Automatic Assignment** When you click the Work button for an open entry, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

> **Note**
>
> **Work List**. This column has a Work List icon. Clicking this icon populates the Work List zone with the results. Clicking on entries in the Work List zone performs all the logic described here when clicking the Work button for one entry. It will drill into the related object or launch a script, if configured. It will make this To Do the 'current' To Do, it will assign it to you if it is **Open** and update it to **In Progress**.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Only 1 **In Progress** To Do entry. If you had a different To Do entry in progress before clicking the Work button, the other To Do entry will be put **On Hold**.

- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Days Old** is the number of days since the To Do entry was created by the system. Entries are sorted from youngest to oldest within priority.
- **Status** is the current status of the To Do entry. If the assignment substatus is populated, that value is displayed, otherwise the To Do entry status is displayed.
- **Assign To User** is the user currently assigned to the To Do entry.
- **Message** is the message associated with the To Do entry.
- **Related** displays a count of entries that are potentially related to the current entry based on having the same foreign key characteristic values. Click the hyperlink to navigate to the To Do Search page with the current entry as the Related To Do Filter value.
- **Note** displays a Comment icon if there are either comments or a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do Entry. Click the icon to open a pop-up window which displays the To Do entry information, message, comments and logs.
- **Characteristics** displays the characteristic types and values for the entry as a concatenated string. By default this list is collapsed to one line to save vertical space. If additional characteristics exist, you can use the expand arrow to display them.

> **Note**
>
> A feature configuration option is supplied to opt out of the compact view. Under the **To Do Options** feature configuration type, define the **Collapse Characteristics Column** option and set the value to "N".

- **Related Entity** displays the information for the object related to this entry. The information string allows you to navigate to the page on which the entry's respective object is displayed.
- **To Do ID** is the system generated key of the entry. Click the hyperlink to navigate to the To Do Entry page to view or maintain the current entry.

Click the **Column Configuration** icon in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- **Key 1** to **Key 5** contain the first five sort key values for the entry.
- **To Do Role** displays the description of the role for the To Do entry.

**Actions**

The zone provides the ability to perform certain actions on one or more selected To Do Entries.

Click the check box in the results list heading to select all entries displayed in the grid or click the box in a row to select a specific entry. The following actions are available:


Copyright © 2026, Oracle and/or its affiliates.

- Use the **Update** button to open a pop-up window in which you can update the priority and/or add a **User Details** log entry to the selected To Do entries.
- Use the **Assign** button to assign a user to the selected To Do entries. Supervisors will be presented with a pop-up window in which they can choose the assignee from a list of users who are members of at least one role for the To Do Type. The user search includes information about the number of To Do entries already assigned to each user. For users without supervisor access, the current user is assumed to be the assignee and the window is not displayed. Note that the system does not allow a non supervisor to assign themselves a To Do that is assigned to another user.
- Use the **Unassign** button to reopen the selected To Do entries and remove the assignee. Note that the system does not allow non supervisors to reopen a To Do that is assigned to another user.
- Use the **Complete** button to complete the selected To Do entries. Both supervisors and non supervisors may complete any To Do entry, including those assigned to other users.

# To Do Dashboard

This portal provides a high level summary of open and assigned To Do entries by various criteria. It is intended to be used by both supervisors and non-supervisor To Do users. An implementation may introduce additional zones to this portal as needed.

> **Note**
>
> **Results are restricted by role.** For supervisors, the charts consider To Do entries that are in a role of which the supervisor is a member. For non supervisors, the charts consider open To Do entries that are in a role of which the user is a member or entries currently assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

Use **Menu**, and then **To Do**, and then **To Do Dashboard** to navigate to the portal.

The following zones may appear as part of the portal's **Main** tab page

- **Filters.** This zone displays and manages filters restricting the information being displayed.
- **To Do Entries By Role.** This zone summarizes To Do entries by their currently assigned role.
- **To Do Entries By Status.** This zone summarizes To Do entries by their current status.
- **To Do Entries By Priority.** This zone summarizes To Do entries by their priority.
- **To Do Entries By Age.** This zone summarizes To Do entries by age based on age ranges defined for the current user.
- **To Do Entries By Type.** This zone summarizes To Do entries by their to do type.

Selecting a specific slice of data on any of these zones sets it as a filter and broadcasts its value to the other zones on the portal allowing them to filter their data accordingly.

The following zone appears as part of the portal's **Details** tab page

- **To Do Details** This zone lists To Do entries that match various criteria and allows the user to work and manage the list.


Copyright © 2026, Oracle and/or its affiliates.

### To Do Details

This zone allows you to search for non-closed To Do entries using various attributes of the entries, and to perform actions on selected ones.

> **Note**
>
> **Results are restricted by role.** For supervisors, this search returns To Do entries that meet the search criteria and are in a role of which the supervisor is a member. For non supervisors, the search returns open To Do entries that meet the search criteria and are in a role of which the user is a member or entries currently assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

The following sections describe the features of this zone.

### Search Filters

The following table describes each of the different filters supported by this zone.

| Search Filter | Description |
|---|---|
| To Do Type | Use this filter to restrict the search to To Do entries of a specific type. Only To Do Types where the user is a member of at least one valid role for the type will be available for selection. If the To Do Role filter is populated, the list is further restricted to valid To Do Types for the To Do Role. |
| Status | Use this filter to restrict the search to To Do entries with a status of **Open** , **Assigned**, **In Progress** or **On Hold**. If you choose Assigned, then it includes the To Do entries that are In Progress and On Hold. Note that the underlying query excludes completed To Do entries. |
| Days Old | On the **To Do Management** portal, this option is represented by a pair of filter fields. Use these filters to restrict the search to To Do entries whose age falls within a given range in days. For example, entering values of 2 - 5 will restrict the search to entries that are greater than or equal to two days old and less than or equal to five days old.<br>On the **To Do Dashboard** portal, this option lists the To Do Summary age ranges as set up for the user. Select a specific age range to restrict the search to To Do entries whose age falls within the given range in days. |
| To Do Role | Use this filter to restrict the search to To Do entries of a specific role. Only To Do Roles where the user is a member will be available for selection. If the To Do Type filter is populated, the list is further restricted to valid To Do Roles for the To Do Type. |
| Assign To User | Use this filter to restrict the search to To Do entries assigned to a particular user. This filter only appears if the current user has To Do supervisor access. |
| Priority | Use this filter to restrict the search to To Do entries with a specific priority. |
| Key Value | Use this filter to restrict the search to To Do entries with a sort key value containing the entered value. |
| Message | Use this filter to restrict the search to To Do entries with a specific message. This filter appears after the To Do Type filter has been populated and shows a list of the message texts from all non closed To Do entries of the selected To Do Type. |


Copyright © 2026, Oracle and/or its affiliates.

| Search Filter | Description |
|---|---|
| Characteristic | Use these filters to restrict the search to To Do entries with a specific **Characteristic Type** and value. Up to five characteristic type and value pairs may be entered. If the To Do Type filter is populated and the To Do Type is 'manual', the list of characteristic types will be restricted to the characteristic types defined for the To Do Type. For automated To Do Types, if valid To Do characteristic types are configured for the To Do type, only those are shown. If an automated To Do Type doesn't define valid characteristics or if no To Do Type filter is populated, all valid characteristic types for the To Do Entry entity are shown.<br>Note that if you have populated characteristic filters and then you change the To Do Type filter, the system retains the characteristic filter as long as the characteristics are applicable to the newly selected To Do Type. |

The system shows the total number of To Do entries displayed and the total number that satisfy your search criteria immediately below the set of results.

> **Note**
>
> For performance reasons, because the filter criteria may be broad, the search zone is configured to only get the first 5000 records that match the criteria and will then sort that number to then display the first 400 sorted by priority and age. The To Do Dashboard portal may be used to get exact counts of records with certain criteria. Use additional criteria to narrow down the search as needed.

### Search Results

The zone will display all To Do entries that meet the search criteria, up to the limit defined. The following columns are displayed initially:

- The Action column includes the button Work. Click this button to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

> **Important**
>
> **Automatic Assignment** When you click the Work button for an open entry, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

> **Note**
>
> **Work List**. This column has a Work List icon. Clicking this icon populates the Work List zone with the results. Clicking on entries in the Work List zone performs all the logic described here when clicking the Work button for one entry. It will drill into the related object or launch a script, if configured. It will make this To Do the 'current' To Do, it will assign it to you if it is **Open** and update it to **In Progress**.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Only 1 **In Progress** To Do entry. If you had a different To Do entry in progress before clicking the Work button, the other To Do entry will be put **On Hold**.

- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Days Old** is the number of days since the To Do entry was created by the system. Entries are sorted from youngest to oldest within priority.
- **Status** is the current status of the To Do entry. If the assignment substatus is populated, that value is displayed, otherwise the To Do entry status is displayed.
- **Assign To User** is the user currently assigned to the To Do entry.
- **Message** is the message associated with the To Do entry.
- **Related** displays a count of entries that are potentially related to the current entry based on having the same foreign key characteristic values. Click the hyperlink to navigate to the To Do Search page with the current entry as the Related To Do Filter value.
- **Note** displays a Comment icon if there are either comments or a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do Entry. Click the icon to open a pop-up window which displays the To Do entry information, message, comments and logs.
- **Characteristics** displays the characteristic types and values for the entry as a concatenated string. By default this list is collapsed to one line to save vertical space. If additional characteristics exist, you can use the expand arrow to display them.

> **Note**
>
> A feature configuration option is supplied to opt out of the compact view. Under the **To Do Options** feature configuration type, define the **Collapse Characteristics Column** option and set the value to "N".

- **Related Entity** displays the information for the object related to this entry. The information string allows you to navigate to the page on which the entry's respective object is displayed.
- **To Do ID** is the system generated key of the entry. Click the hyperlink to navigate to the To Do Entry page to view or maintain the current entry.

Click the **Column Configuration** icon in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- **Key 1** to **Key 5** contain the first five sort key values for the entry.
- **To Do Role** displays the description of the role for the To Do entry.

**Actions**

The zone provides the ability to perform certain actions on one or more selected To Do Entries.

Click the check box in the results list heading to select all entries displayed in the grid or click the box in a row to select a specific entry. The following actions are available:


Copyright © 2026, Oracle and/or its affiliates.

- Use the **Update** button to open a pop-up window in which you can update the priority and/or add a **User Details** log entry to the selected To Do entries.
- Use the **Assign** button to assign a user to the selected To Do entries. Supervisors will be presented with a pop-up window in which they can choose the assignee from a list of users who are members of at least one role for the To Do Type. The user search includes information about the number of To Do entries already assigned to each user. For users without supervisor access, the current user is assumed to be the assignee and the window is not displayed. Note that the system does not allow a non supervisor to assign themselves a To Do that is assigned to another user.
- Use the **Unassign** button to reopen the selected To Do entries and remove the assignee. Note that the system does not allow non supervisors to reopen a To Do that is assigned to another user.
- Use the **Complete** button to complete the selected To Do entries. Both supervisors and non supervisors may complete any To Do entry, including those assigned to other users.

# To Do Entry

Refer to Understanding To Do Processing for detailed explanations about many To Do entry features.

To view a To Do entry, use **Menu**, and then **To Do**, and then **To Do Entry**, and then **Search** . You are brought to a query portal with options for searching for a To Do Entry. Note that some query options limit the results. Refer to the inline help for more information.

> **Note**
>
> **Restricted by role.** Any user may change Comments for any To Do Entry. In addition, any user may add user Log details. However, for all other changes and actions, you may only change To Do entries that are assigned to you or ones that are **Open** and assigned to a role to which you belong. Refer to To Do Entries Reference A Role for more information about roles.

Once a To Do Entry has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The inline help provides information about several fields. The following points highlight information about some of the fields that may be visible.

If the To Do entry has non-complete related entries an insight appears indicating the number of related entries. In addition, details about the related entries are shown in the Related To Do Entries zone.

If the To Do entry was **Created By** a background process, information about that background process appears with a link allowing the user to drill into the batch run, if desired. Refer to To Dos Created for Object-Specific Error Conditions for more information.

If the To Do type is configured to To Do Entries May Be Routed Out Of The System, the **Extract Info** appears. If the extract process is pending, the information indicates that. Otherwise, if it has run, information about the background process appears with a link allowing the user to drill into the batch run, if desired.

The **Characteristics** collection shows existing characteristics. This collection is maintained using the Edit action for users that are allowed to update this To Do entry. Refer to To Do Entry Characteristics for more information.


Copyright © 2026, Oracle and/or its affiliates.

### Actions

Most of the actions are self-explanatory and are related to simply updating the status value. This page has three special actions.

- Using the **Work** button causes several things to occur. You drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. If the To Do entry is Open, it gets assigned to you and the assigned substatus is set to **In Progress**, if your implementation has implemented this functionality. Refer to What Happens When You Work A To Do Entry for more information.
- Use the **Forward** to assign a To Do to another user or role. When a To Do is forwarded, a log entry indicates who forwarded the To Do, to whom it was forwarded, and the specified reason. The dialogue prompts for whether to send to a User or a Role. The search for User includes information about the number of To Do entries already assigned to this user. When choosing a User, confirm the correct Role. The Details provided are captured in the log entry.
- **Send Back** is only applicable if the **Sent By** user is populated. When a To Do is sent back, a log entry is created that indicates when the To Do was sent back and by whom. If the To Do is **Open**, this dialogue allows you to send it back to a different role. If the To Do is assigned to you, the dialogue allows you to re-assign the To Do to the person who assigned it to you. The Details provided are captured in the log entry.

### Related To Do Entries

If the To Do has related To Do entries, this zone appears with information about the related entries. Use this zone to perform actions on the current To Do and its related entries.

- Adding a Manual To Do Entry

## Adding a Manual To Do Entry

If your system is configured with manual To Do types, you can add a To Do entry online.

Open the **Menu**, and then **To Do**, and then **To Do Entry**, and then **Add.**

You are prompted for a **To Do Type**. Only To Do types that are set up for manual usage appear in the list.

The default **To Do Type Priority** is displayed adjacent. Use **Override Priority** to override the default priority assigned by the system.

Type a **Subject** for your To Do entry. Depending on how the To Do type is set up, the subject may appear as the message in the To Do entry and To Do lists.

Enter a **Comment** to provide a longer description of and more details about the To Do entry.

**Send To** indicates whether the new To Do will be assigned to a **Role** (the default) or **User**.

The **Role** indicates the role to which this To Do entry will be assigned. Only users of the specified role will be able to view and work on the To Do entry. The role is defaulted from the To Do type. If you want to assign the To Do to a different role, change the specified role. You can only assign the To Do entry to one of the roles specified on the To Do type.

**User** appears if **Send To** is set to **User**. Use this option if you want to assign the To Do to a specific user. The user must be a member of the selected role.


Copyright © 2026, Oracle and/or its affiliates.

If the To Do type defines valid characteristic types, the **Characteristics** collection is visible. If any of the characteristic types are flagged as Default on the To Do type, the characteristic type with its default value (if applicable) are displayed.

# Legacy UI Features

The pages described in this section are legacy pages that are retained for backward compatibility. The product plans to deprecate the pages described here in an upcoming release.

- To Do Supervisor Functions
- To Do List Functions
- To Do Search

## To Do Supervisor Functions

This section describes several To Do transactions that aid supervisors in the monitoring and assignment of To Do entries.

- Supervisor To Do Summary
- Supervisor User Summary
- Supervisor To Do Assignment

### Supervisor To Do Summary

This Supervisor To Do Summary page summarizes To Do entries by To Do Type. Only entries that are **Open** (i.e., not assigned to a specific user) or **Assigned** are summarized.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Dashboard portal for the page that the product recommends using. Besides providing counts by To Do Type it provides other breakdowns and allows for interaction between graphs.

> **Note**
>
> Only To Do entries that are assigned to a role to which you belong are summarized on this page. Refer to To Do Entries Reference A Role for more information.

Use **Menu**, and then **To Do**, and then **Supervisor To Do Summary** to open this page.

**Description of Page**

This page summarizes all To Do entries that are **Open** or **Assigned**.

> **Note**
>
> This page does not further break down the assigned To Do entries by the substatus. Refer to The Lifecycle Of A To Do Entry for more information on the substatus values.


Copyright © 2026, Oracle and/or its affiliates.

- **To Do Type** contains the description of the To Do type.
- **Priority** provides a total count of To Do entries of a given type by priority. For each distinct priority the description includes the priority value followed by the corresponding number of entries. If all entries are of the same priority no additional count is displayed.
- **Total** contains the total of **Open** and **Assigned** entries. The hyperlink navigates to the Supervisor To Do Assignment transaction, setting it to display all **Open** and **Assigned** entries for the given To Do Type.
- **Open** contains the number of To Do Entries whose status is **Open**. The hyperlink navigates to the Supervisor To Do Assignment transaction, setting it to display all **Open** entries for the given To Do Type.
- **Assigned** contains the number of To Do entries whose status is **Assigned**. The hyperlink navigates to the Supervisor To Do Assignment transaction, setting it to display all **Assigned** entries for the given To Do Type.

The "age bars" that appear to the right of the totals highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is user-configurable.

## Supervisor User Summary

The Supervisor User Summary page summarizes To Do entries by the user who is assigned to work on the entries. Only entries that are **Assigned** are summarized.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for a page that allows you to view To Do entries currently assigned to users.

> **Note**
>
> Only To Do entries that are assigned to a role to which you belong are summarized on this page. Refer to To Do Entries Reference A Role for more information.

Use **Menu**, and then **To Do**, and then **Supervisor User Summary** to open this page.

**Description of Page**

This page contains a grid that displays all assigned To Do entries by User ID and To Do type. It indicates how many To Do entries of the displayed Type are assigned to the user (Assigned this Type) and how many To Do entries in total are assigned to the user (Assigned Total).

> **Note**
>
> This page does not further break down the To Do entries by the substatus. Refer to The Lifecycle Of A To Do Entry for more information on the substatus values.

Clicking the hyperlinks transfer you to the Supervisor To Do Assignment transaction. On this transaction, you can assign / reassign To Do entries to a specific user. The difference is that clicking the To Do Type hyperlink causes only **Open** entries to be displayed on the Assignment


Copyright © 2026, Oracle and/or its affiliates.

transaction. Clicking the count in the Assigned this Type column causes **Assigned** entries assigned to the displayed user to be shown.

## Supervisor To Do Assignment

This page lists To Do entries of a given type and allows a supervisor to assign / reassign them to a designated user.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for the page that the product recommends using. It provides many more filters to narrow down the results. In addition, it provides more data in the results and provides more actions for a supervisor to perform for To Do entries.

Use **Menu**, and then **To Do**, and then **Supervisor To Do Assignment** to open this page.

**Description of Page**

This page contains **Open** or **Assigned** To Do entries of a given type. These entries can be assigned to a specified user one-by-one or several at a time. The following points describe this functionality:

- You can select a different **To Do Type** by typing in a value or pressing the search button in the upper right corner. An **Info** description of the To Do type appears on the left.
- You can change the list of entries displayed according to their status. Select the **Filter by** status to be All (blank), **Open**, or **Assigned**.
- For **Assigned** entries, you may also **Filter by User** to limit the search to a specific user assigned to the To Do entries.
- If you want to re-sort the entries by a different value, click the search button adjacent to **Extra Information**. Each row's respective sort value is displayed in the **Extra Information** column.

> **Note**
>
> **Extra Information.** Each To Do list's default **Extra Information** is defined on the list's To Do Type. Also note that each To Do list's available **Extra Information** differs.

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** to remove the selections in the grid.
- Use the checkbox if you want to assign specific entries.
- Entries are further sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system. Click the Create Date/Time hyperlink to navigate to the To Do page where additional information is visible and where additional actions are possible.
- Click the number of **Related** entries hyperlink to navigate to the To Do Search query to list potentially related entries to the current entry.
- **Worked By** shows the user currently assigned to the To Do entry.


Copyright © 2026, Oracle and/or its affiliates.

- **To Do Entry Status** is the current status of the To Do entry. If the assigned substatus is populated, that value is shown, otherwise the status value is shown. Refer to The Lifecycle Of A To Do Entry for more information.
- **Message** contains a description of the To Do entry. Use the message description hyperlink to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Note that clicking this action does not push the To Do Entry into the Current To Do zone and does not assign the To Do Entry to you. Refer to Working On A To Do for information about using that functionality.
- You can assign entries to a user:
  - Enter a user ID or search for a user using the **Assign To User** field. You can only select a user that is associated with a role that is associated with this **To Do Type**. Note that the current number of assignments is displayed adjacent.
  - Select one or more entries to assign to this user and click the **Assign** button. The entries' status becomes **Assigned**.

## To Do List Functions

This section describes To Do pages that aid users in the assignment and working of entries in To Do lists.

- To Do Summary
- To Do List

### To Do Summary

This page summarizes To Do entries that meet the following qualifications:

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Dashboard portal for the page that the product recommends using. Besides providing counts by To Do Type it provides other breakdowns and allows for interaction between graphs.

- The entry is **Open** and you belong to the entry's To Do role. Refer to To Do Entries Reference A Role for more information about roles.
- The entry is **Assigned** and you are the user to which it is assigned.

Use **Menu**, and then **To Do**, and then **To Do Summary** to open this page.

| Keyboard Shortcut | Alternate Shortcut |
|---|---|
| `Alt+X` | `Alt+Shift+X` |

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


Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture of To Do Lists for an overview of To Do functionality.

**Description of Page**

This page is dedicated to a grid that summarizes the To Do entries on which you can work. The following information is displayed:

- **To Do Type** contains the description of the To Do type.
- **Priority** provides a total count of To Do entries of a given type by priority. For each distinct priority the description includes the priority value followed by the corresponding number of entries. If all entries are of the same priority no additional count is displayed.
- **Total** contains the total of **Open** and **Assigned to You** entries. The hyperlink navigates to the To Do List transaction, setting it to display all **Open** and **Assigned** entries assigned to you for the given To Do Type.
- **Open** contains the number of To Do Entries whose status is **Open**. The hyperlink navigates to the To Do List transaction, setting it to display all **Open** entries for the given To Do Type.
- **Assigned to You** contains the number of To Do entries that are assigned to you whose status is **Assigned**. The hyperlink navigates to the To Do List transaction, setting it to display all **Assigned** entries for the given To Do Type.

> **Note**
>
> This page does not further break down the assigned To Do entries by the substatus. Refer to The Lifecycle Of A To Do Entry for more information on the substatus values.

The "age bars" that appear to the right of the totals highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is user-configurable.

## To Do List

This page lists To Do entries of a given type that are either **Open** (not assigned to a specific user) or are assigned to you.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for a page that the product recommends using. It provides many more filters to narrow down the results. In addition, it provides more data in the results and allows the user to adjust the priority of To Do entries.

> **Note**
>
> **Restricted by role.** Only those entries assigned to a role to which you belong may be seen on this page. Refer to To Do Entries Reference A Role for more information.


Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> We recommend familiarizing yourself with the topics described in The Big Picture of To Do Lists to fully appreciate how To Do entries are managed by the system.

Use **Menu**, and then **To Do**, and then **To Do List** to open this page.

**Description of Page**

This page contains a grid that summarizes the To Do entries of a given type on which you can work.

The **Info** section contains a concatenation of the **To Do Type**'s description and the user's name.

The area beneath **Info** provides you with options that control which To Do entries appear and the order in which they are displayed. The following points describe the various options:

- Use **Sort By** to define how you want the To Do entries sorted. The following options are available:
  - **Assigned Date - On or After.** This option sorts the To Do entries in chronological order (based on the date the entry was assigned to you) starting from the input **Date**. If you choose this option, the system assumes you want to start from the earliest To Do entry. If you want to start from a date other than the earliest To Do entry, enter the desired **Date** and click the adjacent search button.
  - **Assigned Date - On or Before.** This option sorts the To Do entries in reverse chronological order (based on the date the entry was assigned to you) starting from the input **Date**. If you choose this option, the system assumes you want to start from the today's date. If you want to start from a date other than today's date, enter the desired **Date** and click the adjacent search button.
  - **Create Date - On or After.** This option sorts the To Do entries in chronological order (based on the date the entry was created) starting from the input **Date**. If you choose this option, the system assumes you want to start from the earliest To Do entry. If you want to start from a date other than the earliest To Do entry, enter the desired **Date** and click the adjacent search button.
  - **Create Date - On or Before.** This option sorts the To Do entries in reverse chronological order (based on the date the entry was created) starting from the input **Date**. If you choose this option, the system assumes you want to start from the today's date. If you want to start from a date other than today's date, enter the desired **Date** and click the adjacent search button.
  - **Extra Information**. This option sorts the To Do entries by the value of the **Extra Info**.
- Use **Extra Info** to choose data related to the To Do Entry that should be used to sort the To Do entries. This information is displayed in each entry's **Extra Information** column.

> **Note**
>
> **Extra Info.** Each To Do types's available **Extra Info** is defined on the To Do Type Sort Keys. Also note that each To Do type's available **Extra Info** differs based on the configuration of the To Do Type.

- Use **Filter by** to restrict the To Do entries based on their status. The following options are available:


Copyright © 2026, Oracle and/or its affiliates.

- **All**. This option shows all entries that are **Open** or **Assigned**.
- **Assigned**. This option shows all entries that are assigned to you.
- **Open**. This option shows all entries that are **Open**.

The grid that follows contains the To Do entries that match your criteria. The following information appears:

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** button to remove the selections in the grid.
- Use the checkbox if you want to mark specific entries as **Completed** or **Assigned** using the **Complete** or **Assign** buttons at the bottom.
- A Comment icon appears if there are comments on the To Do item and a Log Entry icon appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do item. This column appears if at least one To Do has either comments or a special log entry.
- Entries are further sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- Click the Create Date/Time hyperlink or either of the special icons to navigate to the To Do entry page.
- Click the number of **Related** entries hyperlink to navigate to the To Do Search query to list potentially related entries to the current entry.
- **Assigned Date/Time** contains the date and time the To Do entry was assigned to a user. This field is only populated on entries that are **Assigned**. If a user assigned the To Do entry, the user ID and name of the user who assigned the To Do entry are displayed beneath the Date/Time.
- **Message** contains a description of the To Do entry. Use the message description hyperlink to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

> **Important**
>
> **Automatic Assignment** When you drill into an open entry using the message link, the system changes the status of the entry to **Assigned** / **In Progress** and assigns it to you. It also causes this To Do entry to be your current To Do. Refer to Working On A To Do for more information.

- **To Do Entry Status** is the current status of the To Do entry. If the assigned substatus is populated, that value is shown, otherwise the status value is shown. Refer to The Lifecycle Of A To Do Entry for more information.
- **Extra Information** contains the data for the entry that corresponds to the Extra Info chosen in the sort options above.
- **To Do ID** contains the unique identifier of the To Do entry.
- If you want to mark one or more entries as **Assigned**, check the desired entries and then click the **Assign** button.


Copyright © 2026, Oracle and/or its affiliates.

- If you want to mark one or more entries as **Completed**, check the desired entries and then click the **Complete** button. Completed entries do not appear on the To Do list (but they are retained on the database for audit purposes).

## To Do Search

This page allows you to search for To Do entries using various search criteria, and perform various actions on selected ones. This page includes completed To Do entries in its search results, when applicable with respect to the filter criteria.

Use **Menu**, and then **To Do**, and then **To Do Search** to open this page.

> **Note**
>
> This page will be deprecated in a future release. Refer to the To Do Management portal for a page that the product recommends using. It provides many more filters to narrow down the results. In addition, it provides more data in the results and allows the user to adjust the priority of To Do entries.

> **Note**
>
> **No restriction by role.** This page returns all To Do entries that meet the search criteria, regardless of the To Do entry's role. Using the search, a user may drill into the details of a To Do entry even if the To Do is not assigned to the user. Refer to To Do Entries Reference A Role for more information about roles.

The top half of the page is where you enter the criteria used to search for To Do entries.

> **Caution**
>
> Try to be as specific as possible when entering search criteria. Why? Because entering open-ended search criteria may have a severe impact on response times.

The following table describes each of the different search filters.

| Search Filter | Description |
|---|---|
| Related To Do | If you're searching for potentially related entries for a given To Do entry, choose **Specific To Do** for the **Related To Do Filter**. Enter the **To Do ID**.<br>A filter value of **Not Applicable** defaults. |
| To Do Type | If you're searching for a To Do entry and you know the value of a To Do Type linked to the To Do entry, choose **Specific To Do Type** for the **To Do Type Filter**. Enter the **To Do Type**.<br>A filter value of **All** defaults. |
| Char Type | If you're searching for a To Do entry and you know the value of a characteristic linked to the To Do entry, choose **Specific Char Type** for the **Char Type Filter**. Enter the **Characteristic Type** and **Characteristic Value**.<br>A filter value of **Not Applicable** defaults.<br>Only characteristic types that include a characteristic entity of **To Do Entry** and that have been marked to allow searching by characteristic value are available. |


Copyright © 2026, Oracle and/or its affiliates.

| Search Filter | Description |
|---|---|
| User | Use this search method if you know the **Created By User**, **Assigned To User** or **Completed By User** for the To Do entry. Enter the appropriate **User Id**.<br>A filter value of **Not Applicable** defaults. |
| Date Range | Use this field if you're searching for a To Do entry and know the **Date Created** or **Date Completed**.<br>A filter value of **Date Created** defaults. The **Start Date/Time** defaults to the current date, 12:00AM and the **End Date/Time** defaults to the current date, 11:59PM.<br>If you enter a **Date Range Filter** of **All**, the system ignores this search method. |
| Status | Use this search method to restrict the search to To Do entries with a specific status value. You may search for To Do entries in the status of **Open**, **Assigned** or **Completed**. You may also perform a combination search for To Do entries that have a status value of either **Open** or **Assigned**.<br>A filter value of **All** defaults. |

> **Note**
>
> **The system allows any combination of values.** The system does not try to validate your search combinations, even if they do not make sense. For example, if you enter a **User Filter** value of **Completed By User** and a **Status Filter** value of **Open**, you will not get an error. In this case, the system will find no records that match your criteria (because To Do entries that are Open do not have a Completed By User).

The system shows the total number of To Do entries that satisfy your search results immediately below the grid.

The first group of To Do entries is displayed in the grid at the bottom of the page.

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** to remove the selections in the grid.
- Use the checkbox if you want to perform actions (see below) on specific entries. Only **Open** or **Assigned** entries may be selected.
- A Comment icon appears if there are comments on the To Do item and a Log Entry icon appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do item. This column appears if at least one To Do item has either comments or a special log entry. Use either icon to navigate to the To Do entry page.
- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system. Use the hyperlink to navigate to the To Do entry page.
- Click the number of **Related** entries hyperlink to redisplay this page listing potentially related entries to the current entry.
- **Last Assigned Date/Time** contains the date and time the To Do entry was most recently assigned to a user, if applicable, along with the Assigned To user.
- **Message** contains a description of the To Do entry. If the To Do entry is in the status of **Open** or **Assigned**, the message description allows you to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the


Copyright © 2026, Oracle and/or its affiliates.

respective page, the associated script is launched. Refer to What Happens When You Work A To Do Entry for more information.

> **Note**
>
> **Current To Do**. Drilling into the page by clicking the Message hyperlink does not automatically assign **Open** To Do entries to the user (as the To Do List page does when clicking on the Message). But it does treat this as the user's current to do, enabling the functionality in the Current To Do zone. Refer to Working On A To Do for more information.

- **Status** contains the current status of the To Do entry.
- **Completion Date/Time** contains the date and time the To Do entry was completed and the user who completed the entry, if applicable.
- **To Do Type** contains a description of the To Do type for the To Do entry.

**Actions**

- Use the **Assign** button to assign or reassign the selected entries to a user. Enter a user ID or search for a user using the **Assign To User** field. By default this field is populated with the current user's ID. Note that the user's current number of assignments is displayed adjacent. Upon successful processing, the selected entries are assigned to the specified user and the status of **Open** entries becomes **Assigned**. Additionally, a corresponding log entry is added to each To Do.

> **Note**
>
> You may only assign To Do entries to yourself or to someone else if you are a valid user in the To Do entry's current role.

> **Note**
>
> Only users with **Supervisor Assignment** access rights may reassign entries that are not currently assigned to them.

- Use the **Reopen** button to unassign the selected entries from their users. Upon successful processing, the selected entries become **Open** and a corresponding log entry is added to each To Do.

> **Note**
>
> Only users with **Supervisor Assignment** access rights may unassign entries that are not currently assigned to them and the user must be a valid user in the To Do entry's current role.

- Use the **Complete** button to complete the selected entries.


Copyright © 2026, Oracle and/or its affiliates.

# Reports

The following section describes the functionality provided for submitting reports on-line and for viewing reports submitted through a third party reporting tool.

> **NOT_SUPPORTED**
>
> For information about this functionality, see The Big Picture of Reports.

- On-line Report Submission
- Viewing Historic Reports

## On-line Report Submission

The on-line report submission page enables you to request a specific report to be run. This is only available if your system is able to integrate with your third party reporting system on-line. The system comes configured to integrate with BI Publisher.

- Report Submission - Main

### Report Submission - Main

This page allows you to submit a report on-line. Navigate to this page using **Menu**, and then **Tools**, and then **Report Submission.**.

**Description of Page**

To submit a report, choose the **Report Code** for the process you wish to submit. Refer to Defining Report Definitions for information about defining reports in the system.

A description of the report is displayed.

The parameters defined for the report on the report definition page are displayed in the grid. To see a description of a parameter, use the search button adjacent to the parameter label.

> **Note**
>
> **Default note**. The default values displayed for the parameters are defined on the report definition page.

When you have entered the appropriate parameter values, click the **Submit** button to submit the report. Refer to How To Request Reports for more information.

> **Note**
>
> Reports submitted via the on-line report submission page are not logged in the report history repository and are therefore not visible from the report history page. Only reports submitted through your third party reporting tool's batch scheduler are logged in the report history, if supported by the tool.


Copyright © 2026, Oracle and/or its affiliates.

# Viewing Historic Reports

If your third party reporting tool stores historic copies of reports in a repository, you may configure the system to allow a user to view these historic reports on-line.

> **Note**
>
> This page currently does not display historical reports for BI Publisher.

- Report History - Main

## Report History - Main

This page allows you to view a report stored in your third party reporting tool's report repository, if supported. Navigate to this page using **Menu**, and then **Tools**, and then **Report History**.

**Description of Page**

The top part of the page shows the **Report Code**, **Description** and **Long Description** for the selected report.

To select a report, search by either **Report Code** or **Description** and click the adjacent search button. The reports matching your search criteria are displayed. Choose the desired report from the selection list.

> **Note**
>
> **Displaying Report History**. When you choose a report, the system needs to interface with your reporting software to display a list of historical occurrences of this report.


Copyright © 2026, Oracle and/or its affiliates.

# Part III

# Application Framework Administrative User Guide

Welcome to the Oracle Utilities Application Framework Administrative User Guide. The topics in this guide describe how to administer the Oracle Utilities Application Framework.


- General Options
- Security & User Options
- User Interface Tools
- Database Options
- Configuration Tools
- To Do Lists
- Background Processes
- Attachments
- Monitoring and Analytics
- External Messages
- Integrations
- Market Transaction Management
- Content Migration Assistant (CMA)
- Facts
- Conversion
- Data Export


# General Options

This section describes control tables that are used throughout your product.


- Defining Installation Options
- Support For Different Languages
- Defining Countries
- Defining Currency Codes
- Defining Time Zones
- Defining Geographic Types
- Defining Work Calendar
- Defining Display Profiles
- Defining Phone Types




Copyright © 2026, Oracle and/or its affiliates.

- Defining Feature Configurations
- Defining Master Configurations
- What's New Portal

# Defining Installation Options

The topics in this section describe the various installation options that control various aspects of the system.

- Installation Options - Main
- Installation Options - Messages
- Installation Options - Algorithms
- Installation Options - Accessible Modules
- Installation Options - Installed Products

## Installation Options - Main

Select **Admin**, and then **General**, and then **Installation Options - Framework** to define system wide installation options.

**Description of Page**

The **Environment ID** is a unique universal identifier of this instance of the system. When the system is installed, the environment id is populated with a six digit random number. While it is highly unlikely that multiple installs of the system at a given implementation would have the same environment ID, it is the obligation of the implementers to ensure that the environment ID is unique across all installed product environments.

**System Owner** will be **Customer Modification**.

The **Admin Menu Order** controls how the various control tables are grouped on Admin.

- If you choose **Functional**, each control table appears under a menu item that corresponds with its functional area. Note, the menu that is used when this option is chosen is the one identified with a menu type of **Admin**.
- If you choose **Alphabetical**, each control table appears under a menu item that corresponds with its first letter, using a Roman alphabet. For example, the Language control table will appear under the L menu item entry.

> **Note**
>
> The **Alphabetical** option only supports the Roman alphabet. For languages that do not use the Roman alphabet, the recommendation is to configure the system for the **Functional** setting.


Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> In order to improve response times, installation options are cached the first time they are used after a web server is started. If you change the Admin Menu Order and you don't want to wait for the cache to rebuild, you must clear the cached information so it will be immediately rebuilt using current information. Refer to Caching Overview for information on how to clear the system login cache (this is the cache in which installation options are stored).

The **Language** should be set to the primary language used by the installation. Note that if multiple languages are supported, each user may define their preferred language.

The **Currency Code** is the default currency code for transactions in the product.

If your product supports effective dated characteristics on any of its objects, define the date to be used as the **Characteristic Default Date** on objects without an implicit start date. The date you enter in this field will default when new characteristics are added to these objects (and the default date can be overridden by the user).

**Active Owner** displays the owner of newly added system data (system data is data like algorithm types, zones, To Do types, etc.). This will be **Customer Modification** unless you are working within a development region.

**Country** and **Time Zone** represent the default country and time zone that should be used throughout the application.

> **⚠ Caution**
>
> In most implementations, the time zone defined here matches the database time zone. However, if there is some reason that the database time zone does not match the installation time zone, an implementation may configure a setting in the properties file to automatically convert data from the database time zone to the time zone defined here when displaying dates. Note that when this property setting is defined, changes to the installation time zone will require the server and the thread pool workers to be restarted in order for the changes to take effect.

Turn on **Seasonal Time Shift** if your company requires seasonal time shift information to be defined. Note that this is currently only applicable to Oracle Customer Care and Billing > Interval Billing functionality.

# Installation Options - Messages

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Messages** tab to review or enter messages that will appear throughout the application when a given event occurs.

The **Message** collection contains messages that are used in various parts of the system. For each message, define the **Installation Message Type** and **Installation Message Text**. The following table describes the **Message Types** provided by the framework product and how they are used in the system. Your specific product may have introduced addition message types.


Copyright © 2026, Oracle and/or its affiliates.

| Message Type | How The Message Is Used |
|---|---|
| **Company Title for Reports** | This message appears as a title line on the sample reports provided with the system. Generally it is your company name. It is only used if you have installed reporting functionality and are using the sample reports (or have designed your reports to use this message). |
| **Domain Name** | The text entered for this message type is meant to identify the environment. If populated, the text is displayed next to the product name in The Application Toolbar. It may be used in other cases where it is useful to indicate the environment for a user. For example, the email sent when a background process finishes includes this information, if populated. |

# Installation Options - Algorithms

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Algorithms** tab to review or enter the algorithms that should be evoked when a given event occurs.

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

> ⚠ **Caution**
>
> These algorithms are typically significant processes. The absence of an algorithm might prevent the system from operating correctly.

The following table describes each **System Event**.

| System Event | Optional / Required | Description |
|---|---|---|
| **Validate Email Attachment** | Optional | Algorithms of this type are used to validate the attachments for size and total count while sending attachments using the Email service. Refer to Sending Email for more information |
| **Address Geocoding** | Optional | Algorithms of this type use Oracle Locator to retrieve latitude and longitude coordinates using address information. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| **Attachment Malware Scan** | Optional | Algorithms of this type are invoked when adding attachment content. The algorithm receives the attachment ID and is expected to use a malware scanning tool to verify the contents of the attachment. Only one algorithm is allowed. Refer to Enabling Malware Scan for more information. |
| **Global Context** | Optional | Algorithms of this type are called whenever the value of one of the global context fields is changed. Algorithms of this type are responsible for populating other global context values based on the new value of the field that was changed.<br>Refer to Global Context Overview for more information. |
| **Guaranteed Delivery** | Optional | Algorithms of this type may be called by processes that receive incoming messages that should 'guarantee delivery'. Refer to Guaranteed Delivery for more information. The business service `F1-GuaranteedDelivery` may be used to invoke this plug-in spot. |
| **Ldap Import** | Optional | Algorithms of this type are called for operations on users, groups, and group memberships after they have been processed. |
| **Ldap Import Preprocess** | Optional | Algorithms of this type are called to preprocess data retrieved from LDAP. |
| **Next To Do Assignment** | Optional | This type of algorithm is used to find the next To Do entry a user should work on. It is called from the Current To Do zone when the user asks for the next assignment. Refer to On Demand Assignment Configuration for more information about setting up this information. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| **Reporting Tool** | Optional | If your installation has integrated with a third party reporting tool, you may wish to allow your users to submit reports on-line using report submission or to review report history online. This algorithm is used by the two on-line reporting pages to properly invoke the reporting tool from within the system. |
| **SMS Receive** | Optional | This type of algorithm is used to provide SMS receive service. Only one algorithm of this type should be plugged in. |
| **SMS Send** | Optional | This type of algorithm is used to provide SMS send service. If your installation uses the base algorithm that uses BPEL, you will need to create a feature configuration with the SMS Send Configuration feature type to define your Oracle BPEL server and service call details. If your installation has integrated with a third-party SMS service, you may want to override this algorithm type with your own implementation. Only one algorithm of this type should be plugged in. |
| **To Do Information** | Optional | We use the term To Do information to describe the basic information that appears throughout the system to describe a To Do entry.<br>Plug an algorithm into this spot to override the system default "To Do information". |
| **To Do Pre-creation** | Optional | These types of algorithms are called when a To Do entry is being added. Refer to Refining a To Do Entry's Details at Creation Time for more information. |

# Installation Options - Accessible Modules

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Accessible Modules** tab to view the list of accessible modules.

**Description of Page**


Copyright © 2026, Oracle and/or its affiliates.

This page displays the full list of the application's function modules. A **Turned Off** indication appears adjacent to a module that is not accessible based on your system's module configuration setup.

> **NOT_SUPPORTED**
>
> Refer to Module Configuration for more information on function modules and how to turn off modules that are not applicable to your organization.

# Installation Options - Installed Products

Select **Admin**, and then **General**, and then **Installation Options - Framework** and the **Installed Products** tab to view a read only summary of the products that are installed in the application version that you are logged into.

**Description of Page**

The **Product Name** indicates the name of the "products" that are installed. The collection should include **Framework**, an entry for your specific product and an entry for **Customer Release**.

**Release ID** shows the current release of the application that is installed. This field is used by the system to ensure that the software that executes on your application server is consistent with the release level of the database.

The **Release ID Suffix**, **Build Number** and **Patch Number** further describe the details of your specific product release.

> **Note**
>
> If you have developed implementation-specific transactions, you can populate the release ID, release ID suffix, build number and patch number for the **Customer Release** entry to define the latest version of implementation-specific logic that has been applied to this environment. The framework provides a script (`F1CustRelM`) to allow these fields to be updated manually. In addition, the `F1-InstallationOptions` inbound web service provides operations to view and update those fields via a REST call. Alternatively, your implementation team may populate these fields as part of their upgrade scripts.

The **Product Status** is used to indicate if a product installed by default in the version you are logged into is **Active** or **Inactive**.

The **Display** column indicates the product whose name and release information should be displayed in the title bar. Only one product sets this value to **Yes**.

**Owner** indicates if this entry is owned by the base package or by your implementation (**Customer Modification**).

**Product Type** indicates if the product is a Parallel Application. A parallel application is one that is independent of, and does not conflict with, other parallel applications. Multiple parallel applications can be installed in the same database and application server.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **About Information**. The information on this tab is used to populate the information displayed in the About window for your product.

# Support For Different Languages

- User Language
- Customer Language
- Defining Languages

## User Language

The language English (with the locale "en") is provided by the system and all system metadata are delivered with an English description or label. The system provides support for defining other languages and supports multiple languages in a single environment. System users can use the system in their preferred language, as long as a translation into that language has been provided. A user sees the system in the language defined on their user record. If enabled, users can use the Switch Language zone to switch to another supported language real time.

### Supporting a New Language

Normally, setting up the system for another language (besides the delivered English language entry) is an implementation issue, not an administrative setup issue. However, there are several online administrative features that are used to set up a new language, and these are described here.

1. **Define a language code and indicate that it is enabled.** For details on this procedure, see Defining Languages.
2. **Install or confirm the existence of a language pack**. For an on-premise installation, if the product supplies a language pack with translations for the system metadata descriptions, follow the instructions for downloading the language pack and installing it into your environment. For a cloud installation, if the product supplies a language pack for a given language, no additional step is needed here. The language pack is already installed.
3. **Copy descriptions of all language-enabled tables from an existing translation (e.g., English).** The copied values act as placeholders while the strings are translated into the new language. It is necessary to do this as a first step in order to create records using the new language code created in the previous step. Language-based descriptions can be copied using a supplied batch process, `F1-LANG`. The batch copies all English labels in the system.

   The batch process will also update the new language rows with the translated metadata descriptions from the language pack, if installed.
4. **Translate additional content.** Translatable descriptions and labels for implementation data may be updated / entered in the application. First the user record must be updated to reference the new language. This may be done in one of the following ways:
   a. Switch to the new language using the Switch Language zone.
   b. If that zone is not available, navigate to the user page, assign the new language code to your User ID, sign out, and sign back in again.


Copyright © 2026, Oracle and/or its affiliates.

Any online functions that you access will use your new language code. You can change the language code for all users who plan to use/modify the new language.

> **Note**
>
> The language pack updates all language entries for base owned system data. If your implementation updates base owned labels and descriptions after applying a language pack, they will be overwritten the next time an updated language pack is applied. Note that most user facing labels and messages support defining an Override Label or Override Description. This information is not updated by the base product and should be utilized if your implementation desires a specific label or description.

**Managing Multiple Languages**

Once a non-English language has been set up, there are several points to note when adding or updating records that include translatable strings.

- When a new record is added through the user interface or through a REST service call or any other means that goes through the standard maintenance service, the system automatically generates entries for each supported language and copies the text of the entered description to all the generated languages.
- When the translatable string for a record is updated, only the user's language row is updated. The system does not ripple the change to other supported language entries (because it could be overriding a valid translation).
- When your system is upgraded to a new version, the new version typically includes new system data. The upgrade only includes the English language rows. Implementers should run the `F1-LANG` batch program to generate language rows for the your supported languages for this new data. If a new translation pack for your supported language is available at the time of upgrade, then running this process also applies the translated strings. (As mentioned above, for on-premise installations, you need to download and install the language pack before this step). If the language pack is not available at the time of upgrade, then the batch program should be run again when the language pack has been installed.
- New translatable objects may also be included in a patch / hot fix. In this situation, the `F1-LANG` batch program should be run after installing the patch in order to ensure there are language rows for your supported languages. Updated language packs are only provided for full release upgrades and not for hot fixes. So implementations may need to temporarily supply translations for the new strings until you update to the next release with the next updated translation pack.

# Customer Language

Your specific product may also support capturing the language of a customer. Such that correspondence sent from the product may be produced in a language set on a customer record. Refer to your specific product documentation for more information about additional language support.

# Defining Languages

Your product may support multiple languages. For example, the field labels, input text, and even outputs and reports can be configured to appear in a localized language. A language code for every potential language exists in the system to supply this information in various languages.


Copyright © 2026, Oracle and/or its affiliates.

Select **Admin**, and then **General**, and then **Language** to define a language.

This is a standard All-in-One portal.

When **Language Enable** is checked, the system adds a row for this language whenever a row is added in another language. For example, if you add a new currency code, the system will create language specific record for each language that has been enabled. You would only enable multiple languages if you have users who work in multiple languages. Languages that are configured as enabled, appear in the Switch Language zone.

> **Note**
>
> For on premise applications, the login page displays all the languages that are enabled, allowing the user to toggle the login instructions in that language. The list of enabled languages is captured on the server at startup time. If a new language is enabled, contact your server administrator to refresh the server in order to see the new language displayed in the login page.

The **Locale** is a string containing three portions:

- ISO language code (lower case, required)
- ISO country code (upper case, optional)
- Variant (optional).

Hyphens separate the various portions, and multiple variants are possible.

The locale along with the collator strength control how the contents of grids and search results are sorted by the Java virtual machine (JVM) on your web server: The specific JVM in use by your particular hardware/OS configuration constrains the available **Locales**. Validating the **Locale** against the JVM is outside the scope of this transaction. This means you are responsible for choosing valid **Locales**.

Additionally, external systems that communicate with the product and provide a language use a format that matches the locale. For example in the Accept-Language header for REST web service calls, a requested language is provided.

The following are examples of valid locales:

| Locale | Comments |
|---|---|
| `en-US` | American English |
| `en-AU` | Australian English |
| `pt-BR` | Brazilian Portuguese |
| `fr-FR-EURO` | European French |
| `ja-JP` | Japanese |

The **Collator Strength** parameter is used for the Java collation API . This parameter controls whether, for example, upper and lower-case characters are considered equivalent, or how accented characters are sorted. Valid values for collator strength are **PRIMARY**, **SECONDARY**, **TERTIARY**, and **IDENTICAL**. If you leave this field blank, Java will use its default value for the language. We'd like to stress that the impact of each value depends on the language.


Copyright © 2026, Oracle and/or its affiliates.

Please see https://docs.oracle.com/javase/7/docs/api/java/text/Collator.html for more information about the collator strength for your language.

**Display Order** indicates if this language is written **Left to Right** or **Right to Left**.

Note that all administrative control tables and system metadata that contain language-specific columns (e.g., a description) reference a language code. In addition, other tables may reference the language as a specific column. For example, on the User record you indicate the preferred language of the user.

# Defining Countries

The topics in this section describe how to maintain countries.

- Country - Main
- Country - States

## Country - Main

To add or review Country definitions choose **Admin**, and then **General**, and then **Country.**

The **Main** page is used to customize the fields and field descriptions that will be displayed everywhere addresses are used in the system. This ensures that the all addresses conform to the customary address format and conventions of the particular country you have defined.

**Description of Page**

Enter a unique **Country** and **Description** for the country.

The address fields that appear in the **Main** page are localization options that are used to customize address formats so that they conform to address requirements around the world. By indicating that an address field is **Optional**, you make that field available everywhere addresses for this country are used in the system. You can enter your own descriptions for the labels. These labels will appear wherever addresses are maintained in the system.

> **Note**
>
> Your specific product may also add the ability to mark an address field as **Required**. If that is available, then the product is also supplying appropriate validation in all places where a user defines an address.

> **Note**
>
> For any country where the **State** is enabled, the valid states for the country must be entered on the **Country - State** tab. When entering address constituents on a record that captures this detail, the value for State is verified against the data in the State table. For any country where there is a component of the address that represents a "state" but your implementation does not want to populate the valid states for that country, choose a different field such as County for this constituent (and define an appropriate label). When entering address constituents on a record that captures this detail, no validation is done for the County column.


Copyright © 2026, Oracle and/or its affiliates.

## Country - States

To maintain the states located in a country, choose **Admin**, and then **Country**, and then **Search** and navigate to the **State** page.

**Description of Page**

For any country where you have enabled the State switch, use the **State** collection to define the valid states in the **Country**.

- Enter the standard postal abbreviation for the **State** or province.
- Enter a **Description** for this state or province.

# Defining Currency Codes

The currency page allows you to define display options related to currency codes that are used by your system. Use **Admin**, and then **General**, and then **Currency** to define the currency codes in which financial information is denominated.

This is a standard All-in-One portal.

Most attributes are self explanatory.

> **Note**
>
> Please contact your specific product to verify whether it supports a currency with more than 2 **Decimals**.

The **External Reference** may be used to define a currency whose currency code is more than three characters. To support that type of currency, define a currency code that uniquely identifies the currency in the system and use the external reference to define the currency code recognized by external products. Note that your specific integration with an external system where financial information is included may need to be adjusted to use the external reference instead of the system's currency code. Refer to the specific documentation for your integrations to confirm.

# Defining Time Zones

The following topics describe how to design and set up time zones.

- Designing Time Zones
- Setting Up Time Zones
- Setting Up Seasonal Time Shift

## Designing Time Zones

The Time Zone entity is used to define all the time zones where your customers may operate. Each time zone should define an appropriate Time Zone Name. This is a reference to an external source that defines time zones, their relationship to Greenwich Mean Time, whether the time zone follows any shifting for summer / winter time (daylight savings time) and when this shift occurs.


Copyright © 2026, Oracle and/or its affiliates.

The following sections describe concepts and configuration topics related to managing time zones.

> **Note**
>
> Oracle Utilities Customer Care and Billing - Interval Billing applications customers should consult the topic *Time Issues* (search the Help index for "time issues") for specific information relating to that product's interval billing time related functionality.

### The Base Time Zone

When designing your time zones, the first thing to determine is the base time zone. You may choose the time zone where the company's main office resides. Once this is done you can link the time zone code to the installation option as the base time zone. Refer to Installation Options - Main for more information.

> **Note**
>
> An attribute in the system properties file may be configured to indicate that the DB session time zone should be synchronized with the value defined on the Installation Options. Refer to the *Server Administration Guide* for more information.

If your company does business beyond your main office's time zone, define the other time zones where you may have customers or other systems with which you exchange data. At this point, your specific product may include configuration tables to capture default time zones, for example based on a postal code or geographic location.

### Standard vs. Legal Time

The term **Legal Time** refers to the actual clock time as may be impacted by daylight saving (also called 'local time'). This time line has a gap hour on entry to the daylight saving period and a duplicate hour on exit from it.

The term **Standard Time** refers to a time line without any daylight saving shifts applied to it. This virtual time line has no gap hour on entry to the daylight saving period nor a duplicate hour on exit, i.e. it is continuous and non-ambiguous. As a general rule, it is recommended that all time sensitive data is stored in the standard time of the base time zone as defined on the installation options. This will prevent any confusion when analyzing data and will ensure that your algorithms do not have to perform any time zone or daylight saving shifting of data that may be stored in different time zones.

Depending on your specific product, entities may have their date/time information stored in either of the following options:

- Standard time of the base time zone.
- Standard time of another time zone related to the entity’s geographical location.
- Legal time of the base time zone.
- Legal time of another time zone related to the entity’s geographical location.

The metadata definition of a table specifies whether any of its date/time fields are stored in standard time or not. While typically all date/time fields of a table that supports standard time would be stored in the same way the system allows each field to specify either of these options.


Copyright © 2026, Oracle and/or its affiliates.

- **Physical Standard Time** indicates that the field is stored in base standard time and is pre-populated by the system. For example, creation time, status update time etc.
- **Logical Standard Time** indicate that the field is stored in base standard time.
- **Referenced Time Zone** indicate that the field is stored in standard time of a time zone related to the entity's geographical location.

### Storage vs. Display

Regardless of the time zone option used for storing the data in standard time, all date/time information is entered and displayed on the user interface in the legal time of its respective time zone. If a field is defined to be stored in the standard time of the base or other time zone, the user interface is designed to accept the data in the legal time of the respective time zone and shift it to the standard time of the same time zone before saving the data and performs the reverse shift from standard time back to legal time before displaying the data to the user.

It is important to understand that time zone and daylight saving time conversions to and from standard time are performed behind the scenes as part of the user interface layer. Once data reaches the server it is already in its storage time option. The user should not be aware of these time conversions.

Time shifting between standard and legal times does not happen automatically. When dealing with fields stored in standard time, their data entry and display need to explicitly include time shifting features as part of their user interface and zone configuration.

### Date / Time Schema Elements

When defining date / time fields in a business object schema, schema attributes can be used to define whether or not data should be stored in standard or legal time for the base time zone or if it should be stored in the standard time or legal time of another time zone (related to the data).

By default data stored in standard or legal time is displayed in the corresponding time zone's legal time. An additional schema attribute can be used to indicate if the display of the time should be shifted to the legal time of a different time zone. For example, if the data is stored in the base time zone but the data is related to a different time zone, the data will be shown in the time zone appropriate for the data (including the appropriate seasonal adjustment). Refer to Schema Nodes and Attributes- Standard Time Considerations for more information.

### Exchanging Date / Time Information With External Systems

Date/time information should be exchanged between systems in standard XSD format as it includes a reference to the time zone of the specified time as an offset from Coordinated Universal Time (UTC). The system automatically converts date/time elements in an inbound message from XSD to internal format and vice versa for an outbound message. The latter is controlled by an explicit date/time format setting on the External System record for the outbound message type.

The offset is determined as followed and it is based on the time zone associated with the element as explicitly defined by the respective inbound or outbound schema:

- Elements defined as stored in the standard time of the base or other time zone would always have the same offset throughout the year as this time line is never shifted for daylight saving. The offset would be the standard offset of that time zone from UTC, i.e. not during daylight saving time.
- Elements stored in the legal time of the base time zone or other time zone would have the standard offset of the their storage time zone when the date/time value is outside the daylight saving period and the shifted offset when it is inside that period of time.


Copyright © 2026, Oracle and/or its affiliates.

**User Time Zone**

If your company does business in multiple time zones, the user record of each user may reference the time zone of their location. This time zone provides additional information about the location of the user and as such may be used by specific business rules that involve time zone settings.

> **Note**
>
> The time zone defined on the user record is not used to automatically display date/time information in the user's time zone. By default, the system displays date/time fields in the legal time of their respective time zone and not the user's time zone.

## Setting Up Time Zones

Refer to Designing Time Zones for background information about defining time zones.

Open this page using **Admin**, and then **General**, and then **Time Zone**, and then **Search** .

This is a standard All-in-One portal.

Refer to the embedded help text for more information on the Time Zone attributes.

## Setting Up Seasonal Time Shift

> **Note**
>
> The information in this topic applies only to Oracle Utilities Customer Care and Billing - Interval Billing applications.

Open **Admin**, and then **General**, and then **Seasonal Time Shift**, and then **Search** to define the seasonal time shift schedule.

**Description of Page**

Enter a unique **Seasonal Time Shift** code and **Description** for the seasonal time shift.

The Collection defines the **Effective Date/Time** (in standard time) that a time zone may shift in and out of standard time. If time is changed from standard time on the effective date/time, enter the **Shift in Minutes** that the time changes from standard time (usually `60`). If the time is changed back to standard time on the effective date/time, enter a **Shift in Minutes** of `0`.

# Defining Geographic Types

If your company uses geographic coordinates for dispatching or geographic information system integration, you need to setup a geographic (coordinate) type for each type of geographic coordinate you capture on your premises and/or service points (geographic coordinates can be defined on both premises and service points).


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Product specific.** There is no framework functionality that uses this information. Refer to your specific product documentation to verify how this table is used in your specific product. In addition, you can determine if this object is a foreign key on any tables specific to your product. Navigate to the Table page in the application, search for and select the Geographic Type table and use the Relationships tab to view this information.

To define geographic types, open **Admin**, and then **Geographic**, and then **Geographic Type.**

This is a standard All-in-One portal.

The **Validation Format Algorithm**, if specified, is used to validate that the geographic location entered on the premise and/or service point for the geographic type is in the format as defined in the algorithm. If you require validation, you must set up this algorithm in the system.

# Defining Work Calendar

This portal is used to view and maintain information about a work calendar. Work calendars are used to ensure system-calculated dates fall on a workday.

You can access the portal from **Admin**, and then **General**, and then **Work Calendar**. You are brought to a query portal with options for searching for a specific calendar. Once a calendar has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones will appear as part of the portal's **Main** tab page

- **Work Calendar.** This zone provides information about the selected calendar.
- **Work Calendar Holidays.** This zone lists the holidays observed by your organization for this calendar. Use the zone actions to add, maintain or duplicate a work holiday entry.

# Defining Display Profiles

When you set up your users, you reference a display profile. A user's display profile controls how dates, times, and numbers displayed. Choose **Admin**, and then **General**, and then **Display Profile**, and then **Search** to maintain display profiles.

**Description of Page**

Enter a unique **Display Profile** and **Description** to identify the profile.

Enter a **Date Format**. This affects how users view dates and how entered dates are parsed. The following table highlights standard supported date mnemonics and what is displayed at runtime.

| Mnemonic | Comments |
|---|---|
| dd | Day of the month. |
| d | Day of the month, suppressing the leading 0. |
| MM | Month number. |
| M | Month number, suppressing the leading 0. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Comments |
|---|---|
| yyyy | The 4-digit year. |
| yy | The 2-digit year. |
| y | Allows entry in either 2 or 4-digit form and is displayed in 2-digit form. |

Other characters are displayed as entered. Typically, these other characters should be separators, such as "-", ".", or "/". Separators are optional; a blank space cannot be use.

Examples:

| Configuration Format | Sample Output |
|---|---|
| MM-dd-yyyy | 04-09-2001 |
| d/M/yyyy | 9/4/2001 |
| yy.MM.dd | 01.04.09 |
| MM-dd-y | 04-09-01 - In this case you could also enter the date as 04-09-2001 |

> **Note**
>
> For centuries, the default pivot for 2-digit years is 80. Entry of a 2-digit year greater than or equal to 80 results in the year being interpreted as 19xx. Entry of a 2-digit year less than 80 results in the year being interpreted as 20xx.

In addition, the following date localization functionality is supported. Note that in every case, the date is stored in the database using the Gregorian format. The settings below result in a conversion of the date for the user interface.

- **Hijri Dates**

  Entering **iiii** for the year is interpreted as a year entered and displayed in Hijri format. For example, the Gregorian date 2014–05–30 may be entered / displayed as 1435/07/30 for a user whose display profile date format is iiii/MM/dd. Note that this functionality relies on date mapping to be defined in the Hijri to Gregorian Date Mapping master configuration. entry. Refer to Additional Hijri Date Configuration for more information.

- **Taiwanese Dates**

  Entering **tttt** for the year is interpreted as a year entered and displayed in Taiwanese format where year 1911 is considered year 0000. For example, if the Gregorian date is 01-01-2005, it is displayed as 01-01-0094 for a user whose display profile date format is dd-mm-tttt.

- **Japanese Dates**

  There are two options available for configuring Japanese Era date support. The setting **Gyy** for the year is interpreted as a year entered and displayed using an English character for the era followed by the era number. The letter 'T' is used for dates that fall within the *Taisho* era. The letter 'S' is used for dates that fall within the *Showa* era. The letter 'H' is used for dates that fall within the *Heisei* era. The letter 'R' is used for dates that fall within the *Reiwa* era. For example, for a user whose display profile date format is Gyy/mm/dd the Gregorian date 2008/01/01 is shown as H20/01/01; the Gregorian date 1986/03/15 is


Copyright © 2026, Oracle and/or its affiliates.

shown as S61/03/15. The setting **GGGGyy** is interpreted as a year entered and displayed using Japanese characters for the era followed by the era number.

Japanese date limitations are as follows:

- The years 1912 through the current date are supported.
- Any functionality that displays Month and Year does not support Japanese Era dates. These dates are shown in Gregorian format.
- Graphs that display dates do not support the **GGGGyy** format.

Enter a **Time Format**. The following table highlights standard supported date mnemonics.

| Mnemonic | Comments |
|---|---|
| hh | The hour 1-12. |
| h | The hour 1-12, suppressing the leading 0. |
| HH | The hour 0-23. |
| H | The hour 0-23, suppressing the leading 0. |
| KK | The hour 0-11. |
| K | The hour 0-11, suppressing the leading 0. |
| kk | The hour 1-24. |
| k | The hour 1-24, suppressing the leading 0. |
| mm | Minutes. |
| m | Minutes, suppressing the leading 0. |
| ss | Seconds. |
| s | Seconds, suppressing the leading 0. |
| a | Indicates to include **am** or **pm**. This is only needed for 12 hour formats, not 24 hour formats. (**hh**, h, KK, K). If an **am** or **pm** is not entered, it defaults to **am**. |

Examples:

| Configuration Format | Sample Output |
|---|---|
| hh:mma | 09:34PM (can be entered as 09:34p) |
| hh:mm:ss | 21:34:00 |
| h:m:s | 9:34:0 |

There are several options for displaying Numbers.

**Decimal Symbol** defines the separator between the integer and decimal parts of a number. Valid values are "." (a period) or "," (a comma).

**Group Symbol** defines the means to separate groups of bigger numbers. Valid values are as follows:

- A comma (","). Large numbers group by threes separated by a comma, for example 1,000,000.
- A period ("."). Large numbers group by threes separated by a period, for example 1.000.000.
- **None**. Large numbers do not have any separator, for example 1000000.


Copyright © 2026, Oracle and/or its affiliates.

- **South Asian**. This option uses a comma for its separator but will group large numbers as follows: the first comma is used for the thousands separation and numbers over 9,999 are grouped with 2 units, for example 10,00,000.
- **Space**. Large numbers group by threes separated by a space, for example 1 000 000.

**Negative Format** defines how negative values are displayed. Valid values are `-9.9`, `(9.9)`, or `9.9-`.

**Currency** values can have a different **Negative Format** from other numbers. Valid values are `-S9.9`, `(S9.9)`, or `S9.9-`, where the "S" represents the currency symbol.

- Additional Hijri Date Configuration

## Additional Hijri Date Configuration

For implementations that wish to support displaying dates according to the Hijri calendar, besides appropriate configuration in the Display Profile, the mapping between the Hijri dates and the Gregorian dates must be entered. This mapping is defined in the **Hijri to Gregorian Date Mapping** master configuration record.

The mapping record contains a collection of entries for each year in the Islamic calendar.

For each year, clicking the Expand Zone icon shows the mapping collection with the first date of each month of the Hijri calendar. The corresponding date in the Gregorian calendar should be entered for each row.

# Defining Phone Types

Phone types define the format for entering and displaying phone numbers.

To add or review phone types, choose **Admin**, and then **General**, and then **Phone Type.**

This is a standard All-in-One portal.

The **Phone Number Format Algorithm** for each phone type controls the format for entry and display of phone numbers.

Use **Phone Type Flag** is used to identify phone types that may have specific functionality associated with it. For example if your product had support for sending a facsimile transmittal, identify the phone type that captures a **Fax** number.

# Defining Feature Configurations

Some system features are configured by populating options on a feature configuration.

You can access the portal from the **Admin Menu**, and then **General**, and then **Feature Configuration**. You are brought to a query portal with options for searching for a specific feature configuration. Once a feature configuration has been selected you are brought to the maintenance portal to view and maintain the selected record.

Because various options throughout the system may be controlled by settings in feature configuration, this section does not document all the disparate possible options. The administration documentation for each feature describes its options and whether an option supports multiple values. You may also refer to the detailed description of each option type when reviewing a configuration online for more information.

You can create options to control features that you develop for your implementation. To do this:


Copyright © 2026, Oracle and/or its affiliates.

- Review the lookup values for the lookup field EXT_SYS_TYP_FLG. If your new option can be logically categorized within an existing feature type, note the lookup value. If your new option warrants a new feature type, add a lookup value to this lookup field.
- Define the feature's option types. If you have identified an existing feature type to add the options to, find the lookup with the name xxxx_OPT_TYP_FLG where **xxxx** is the lookup value of EXT_SYS_TYP_FLG noted above. If you decided to create a new feature type (by adding a new lookup value to the EXT_SYS_TYP_FLG lookup, you must create a new lookup with the name xxxx_OPT_TYP_FLG where **xxxx** is the new value you defined above.
- Flush all caches.

The following zones may appear as part of the portal's **Main** tab page.

- **Feature Configuration.** This zone provides information about the selected feature configuration.
- **Options.** This zone is used to maintain various options associated with the feature type.
- **External System Messages.** If the feature exists to interface with an external system, you can use this zone to define the mapping between error and warning codes in the external system and our system. The zone is visible for feature types that their definition in the **Feature Type Configuration** Extendable Lookup supports this functionality.

> **Note**
>
> A corresponding message must be defined in the system message tables using a Message Category that is 90000 or greater so as to avoid collisions during upgrades.

# Defining Master Configurations

A master configuration is an object that enables an implementation to define configuration for features in the system. It is an alternative to using feature configuration for defining options. A master configuration is defined using a business object. Typically one master configuration exists for a given business object yet some configurations may be designed to support multiple records.

### Overview / Initial Setup

The product provides one or more master configuration that may be used for configuration. Some examples of base master configuration business objects are as follows

- **Hijri to Gregorian Date Mapping**. This allows an implementation that uses Hijri dates to define the mapping between those dates and Gregorian dates.
- **ILM Configuration.** For implementations that use Information Lifecycle Management, the ILM configuration record defines some parameters used by the process.
- **Migration Assistant Configuration.** For implementations that use the content migration assistant (CMA), the configuration record defines some parameters used by the process.

For a list of all the master configuration records provided by the product, navigate to the master configuration page in the application. To find help topics related to functionality controlled by the master configuration records, use the keyword ‘master configuration’ in the index.

Implementations may configure the business objects to limit the ones that are visible to users if desired.


Copyright © 2026, Oracle and/or its affiliates.

- If the master configuration does not apply to your implementation, navigate to the business object for this master configuration and update the Instance Control to **Do not allow new instances**. This ensures that the business object will not be visible to any user for the implementation.
- If a given master configuration record is only relevant for certain users in the system, application security may be used to limit the records visible by a given user. By default the base delivered business objects typically reference the Master Configuration MO application service. This may be overridden to link custom application services to the master configuration BOs to provide more granular security. Users will only see the master configurations that they have security for.

### Configuration

You can access the portal from the **Admin**, and then **General**, and then **Master Configuration**. You are brought to a query portal with options for searching for a master configuration business object. If a configuration has not yet been set up for the business object use the add icon to define the record. If one or more configuration records exists for the business object you may select the master configuration business object. Once selected you are brought to this portal to view the selected record.

The following zones may appear as part of the portal's **Main** tab page.

- **Master Configuration** This zone provides information about the selected master configuration record. Please see the zone's help text for information about this zone's fields.
- **Related Configurations** This zone lists additional master configuration records for the same business object, representing different parts of the same configuration. You may add additional parts from this zone. The zone is only visible when the business object is designed to support multiple configuration parts and is explicitly marked as such by the **Multiple Configurations** business option.

# What's New Portal

> **Important**
>
> This is only available for cloud implementations.

The What's New portal provides a list of maintenance pack fixes that have been applied to your environment along with a summary of the maintenance packs that have been supplied along with their dates. There is also a link for Release Notes. Clicking this link opens the Cloud Application Readiness page. Select the link corresponding to your product and version to see the new features added in your version.

To view the information, open the Help Menu in the toolbar and choose the What's New option.

The following zones appear on the Main page.

- **Maintenance Pack Details**. This zone is populated whenever a maintenance pack or a hot fix is applied to your environment. By default it shows fixes applied in your current maintenance pack and it suppresses copies if your environment already received this fix in a previous release. Use the filters to expand the search to other releases or to include copies. There are also filters to narrow the search based on various criteria. Refer to the zone inline help for more information.


Copyright © 2026, Oracle and/or its affiliates.

- **Installation History Summary**. This zone is visible if you click the 'View Installation Summary' link in the previous zone. It shows a distinct list of installation date and version information.

> **Note**
>
> This zone only has fixes that were applied to your environment after the enhancement to populate this data was fully supported. The system is not populating this zone with fixes applied to your implementation prior to the enhancement.

# Security & User Options

The contents of this section describe how to maintain a user's access rights.

- Application Security
- Row Security
- Data Privacy
- Cryptography Keys

## Application Security

The system restricts access to various components of the application using application services. The following sections provide more information about this mechanism.

- Understanding Application Security
- Defining Application Services
- Defining Security Types
- Defining User Groups
- Defining Users
- Application Security Query

### Understanding Application Security

The contents of this section and supporting topics provide information about application security.

#### Users

Each user who is granted access to the application is stored in the User object. This object defines the attributes of the user including identifier, name, language, time zone, home page and display profile (which defines the preferred format of dates and amounts). In addition there is configuration related to using the product, including favorite links and scripts, portal preferences and valid To Do roles - controlling what types of To Do entries a user can work on. Additionally, Refer to Users and User Groups for additional information.


Copyright © 2026, Oracle and/or its affiliates.

### Application Services

Each object in the system that may be secured is associated with an application service. For example, each page in the system defines its application service allowing you to control what users can view a given page.

An application service defines one or more Acces Modes, which typically correspond to an action associated with the application service. This allows you to control users that can only view a type of record (and grant security to the Inquire access mode) and users that can perform other actions such as add, change and delete. Refer to Secured Objects and Application Servicesto understand what objects are associated with an application service and the typical access modes associated with each type of secured object.
Application Services may also define additional configuration, called security types. Refer to Field Level Security for more information.

### User Groups

User Groups are a mechanism for grouping users, usually around job roles. Users are attached to User Groups through a relationship that is effective-dated. This can be useful for the attachment of temporary employees such as contractors or for people who change roles regularly.

Each User Group is authorized to access appropriate Application Services and their access modes. In addition, user groups define access to the application service's security types, if applicable. Refer to Field Level Security for more information.

- Users and User Groups
- Secured Objects and Application Services
- Field Level Security
- Importing Security Configuration from an External Source

## Users and User Groups

A user group defines the application services and access modes its users have access to. A user may be associated to more than one user group. The user groups may have overlapping security configurations and the security access for a user is the union of all security granted by all of its user groups.

A major effort for implementations is to decide how to define their user groups. User groups can be created to represent roles or other job-related categorization; or based on organizational factors; or a combination of both. Refer to Secured Objects and Application Services for some guidelines on understanding what application services secure and tools provided to help security administrators identify what application services to grant access to.

The following sections provide additional topics related to users and user groups, including some base functionality related to these objects as well as some best practices.

### Base Delivered User / User Groups

The system is delivered with a base User linked to a base User Group, provided to facilitate an initial user accessing the application for an on-premises implementation.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Cloud implementations should refer to the Cloud Services Administrative Guide for information on setting up users.

The system supplied user is **SYSUSER**. This user is delivered so that when an on-premises environment is provisioned, there is at least one user authorized to use the system. It has access to the base delivered user groups. An initial administrator can sign in as this user and proceed to create the actual users that should get access to the environment.

> **Important**
>
> Once appropriate users have been defined, it is recommended to update SYSUSER to set it to disabled. (In cloud implementations, this user is delivered as disabled.)

The product provides several base user groups that serve as a kind of library of services. They are meant for documentation purposes and as a way to easily access the system in a development or testing environment. Some user groups are meant to define a list of services associated with a distinct set of features or functions in the system. The user group `ALL_SERVICES` is a catch-all user group that defines services that are not part of a specific use case. (Note that some services may be defined in ALL_SERVICES as well as another specific user group). You cannot change or remove the information delivered for the base user groups. System administrators should define user groups appropriate for their business requirements.

> **Important**
>
> It is not recommended for any production user to have access to the base delivered user groups that are provided as a library of services. The product often adds or changes records in the base user groups for a new release when there is new or changed functionality. It means that your production users would have a change in what they have access to without the ability to opt in to new features.

### Defining a Read-Only User Group

The product does not supply a user group that is 'read only', but it is a common requirement for implementations. This section provides a guideline for creating such a user group.

1. Create the User Group with the desired name and description.
2. Navigate to the Service Manager tab. In the Add Application Service / Access Modes tab, you can use the filter to limit the results to those with Inquire access mode. Click the checkbox in the header to select all rows and click Add. You are prompted for an expiration date. Click Save. (Note that the zone has a limit to 1000 services so you may need to do this more than once to get all services added.)
3. In addition, you should give this user group access to all the access modes for the default application service: `F1-DFLTS` (Default Application Service). This application service is considered one that all users should have for basic access to the system.


Copyright © 2026, Oracle and/or its affiliates.

### Template User

The system provides support for defining one or more users as a template user. There are two broad use cases for using a template user.

#### Inheriting Preferences or Favorites Configuration

Some implementations want all users of a certain team to have the same configuration of portal preferences and / or favorite links. This allows users in that team to have the same information in the same place on their respective displays, facilitating in helping each other with questions.

If your organization wants to implement this practice, define a template user with the desired portal preferences and / or favorite links. You may define different users with different configurations for more granularity.

For each user whose preferences or links should be taken from the template user, go to the User record via **Admin**, and then **Security**, and then **User.** If the portal preferences should be inherited from a template user, define the desired template user as the **Portals Profile User** . If the favorite links should be inherited from a template user, define the desired template user as the **Favorites Profile User**.

If applied, the specified user's portal preferences and / or favorite links will be those defined on the template user and they will be unable to customize those preferences.

#### Defining a Template User to help in User Creation

The user object in the product captures configuration related to many aspects of using the product. It includes application security configuration, such as its user groups; row level security configuration through its access groups; the To Do roles that define what types of To Do entries a user can work on, along with other configuration. Many users that are in a similar 'role' in the organization would have many of the same settings. As such, it may be useful for implementations to define a template user that represents a 'role' in the organization, with much of the configuration common for users in that role. When creating a new user, you can use the template user as the starting point and then duplicate that record to create the new user. Then you only need to adjust the information that is unique for the newly created user.

Many implementations provision their users first in an external identity management system and use an interface to import the user definition into the product. Refer to Importing Security Configuration from an External Source for more information. For these interfaces, the external system may have the ability to define custom information relevant to the product's user record, but the product also supports referencing a template user so that information from the template user is copied to the new user while creating the record.

How can you refer to a template user when creating a User from and external source?

- When the new user is uploaded to the system, the interface uses the user BO `F1-IDMUser` to create the user. The BO includes a pre-processing algorithm. This algorithm looks for the existence of a template user in the schema of the new user sent as a characteristic of type `F1-TMUSR`. If such a characteristic is found, it uses that template user. If not found, the algorithm supports defining a template user as an algorithm parameter. If a template user is identified, all the information from the template user will be copied onto the new user record except for the information passed in from the external source. The template user is captured on the newly created user via a characteristic for information / audit purposes.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Bookmarks** are not included in the data that is copied from a template user.

- Once the new user is created, its configuration can now be adjusted, if applicable. Note that the template user is only a tool used when adding a user. Updates to the template user will not "ripple" to all the other users that were created based on this template.
- Not all external sources may have the ability to define a template user and send it as a characteristic or via another field. For these integrations, using a single template user on the algorithm as a parameter may be used to define basic configuration applicable to a large percentage of new users. (More information on this below). For those integrations that are able to associate a new user with a template user in the external system and provide that template user along with the new user information allows you to associate the new user with a template user that is more aligned with the user's role. Refer to Importing Security Configuration from an External Source for more information on how each integration supports the definition of a template user.

**Designing your Template Users for helping Create Users**

Before configuring template users, all the administrative control tables that are part of User configuration must be defined, including time zones, display profiles, To Do roles, data access roles and user groups.

If your integration will use a single template user defined as an algorithm parameter, you need to define a template user with the basic information that is applicable to all or at least most users. See below for guidance on referencing this template user as an algorithm parameter.

If the way you create users allows you to associate the user with a more specific template user, the next step is to define the user configuration for your system users. During this exercise, you will find that you have broad categories of users. But you will also see that within a given category of user there may be variations in the user privileges and preferences. For example, perhaps there are supervisors within the Customer Service role that have more security privileges than a typical Customer Service user. In addition, there may be variations based on the attributes of the users themselves. For example, maybe your organization exists in multiple time zones and some of your workers are in one time zone and some are in the other.

At this point your security users that are designing their user provisioning procedures must decide the following:

- What information about a new user will be captured in the identity system (besides the expected information like Name, Email and the User IDs)? For example, for the case of multiple time zones, maybe the best solution is to capture the time zone when defining the user.

  What information is defined on the template user and how many template users should be created to reduce the need for manual steps or additional data captured in the identity management system? In the case of multiple time zones, proliferating the template users to have one set for one time zone and another set for the other time zone may not make sense since this is one field that is different. However it may be reasonable to create additional templates in the case of striations in the levels of privileges for workers of a different category. So rather than template users for Dispatcher, Mobile Worker, System Administrator and Contractor, your organization may have template users for Dispatcher, Mobile Worker, Mobile Worker (Supervisor), System Administrator, Contractor (Short Term) and Contractor (Long Term).

  What information is must be configured on the User record in the application after the user is added? If only a small number of users have a variation from other users, it may be that


Copyright © 2026, Oracle and/or its affiliates.

the easiest way to deal with those variations is to simply update those user records manually. Using the above examples:

- If your organization covers 2 time zones but only a small group of people work in one of the time zones whereas the bulk of the users are in the other time zone, the simplest procedure may be to define the template users for the main time zone and use that for the creation of all users. Then for the small group of users in the separate time zone, navigate to the User page to adjust the time zone after the record is added.
- If only a small number of Mobile Workers are supervisors with separate privileges, rather than defining a special template user for those type of workers, the simpler procedure may be to use the Mobile Worker template and then navigate to the User page to add the additional privileges to the supervisor users after the record is added.

**Defining a Template User**

Regardless of which use case is applicable for your organization, the procedure for defining a template user is the same.

Navigate using **Admin**, and then **Security**, and then **User.**

- Define a User ID. If you are using a template user to help create users from an external system and the external system is able to capture the template user to send in with the new user information, this is the ID that should be referenced in the external system.
- Be sure to choose a **User Type** of **Template User**.
- Define all the information appropriate based on the use case.
  - For template users used as a Portal Profile user, be sure to define the appropriate portal profile information.
  - For template users used as a Favorites Profile user, be sure to define the appropriate favorite links.
  - For template users acting as a type of 'role' to help with creation of users, define all the information that should be copied onto a new user that references this user as its template user. (Note that **Bookmarks** are not included in the data that is copied from a template user.)

**Defining a Default Template User for Integrations**

Define the default template user on the algorithm that defaults data from this user as follows:

- Navigate using **Admin**, and then **System**, and then **Algorithm.**
- Search for and select the algorithm `F1-OIMUSR` (Populate User Data from a "Template" User).
- Edit the record. Add a new row in the Algorithm Versions collection. Enter an appropriate date (later than the date of the existing entry, but on or before the current date). Populate the template user you want to use as the default.

## Secured Objects and Application Services

The system restricts access to transactions or explicit services using an application service. The following sections provide more information about what objects are secured, how to find out what is secured by each app service as well as additional information related to managing application services.


Copyright © 2026, Oracle and/or its affiliates.

### What is Secured by an Application Service

- The following points highlight security related to viewing and modifying individual records in the system:
  - All maintenance objects define an application service that includes the basic actions available, typically **Add**, **Change**, **Delete**, and **Inquire**. The base product supplies an application service for every maintenance object. Some objects that include a fixed lifecycle, may define additional access modes on the maintenance object's application service. The application service for the maintenance object is defined on its related service program.

    > ⚠ **Caution**
    >
    > Important! If an application service supports actions that modify the database other than **Add**, **Change**, and **Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a maintenance object that includes access modes **Freeze**, **Complete** and **Cancel**). If you want to give a user access to any of these special actions, you must also give the user access to the **Change** action.

  - For maintenance objects whose user interface page is not portal-based, the application service also controls whether the menu entry appears. If a user doesn't have access to the maintenance object's application service, the menu item that corresponds with the application service will not be visible.
  - For portal based user interfaces, each main (stand-alone) portal defines an explicit application service with the access mode **Inquire**, allowing the user interface to be secured independently of the underlying object security. If a user doesn't have access to the portal's application service, the menu item that corresponds with the application service will not be visible. The base product supplies an application service for every portal that is accessible from the menu. Note that the application service for the portal is defined on its related service program, which is derived via its navigation option and navigation key.
  - Menu items may define an application service / access mode. Typically the security supplied for portals and maintenance objects provides enough granularity to suppress menu items that a user does not have access to. In addition, by default, the system populates application services / access modes on menu items related to a BPA script. A common use case is the 'Add' menu item for maintaining an object. In this case, typically the application service for the maintenance object is used along with the **Add** access mode. Other types of BPA related menu items may have other appropriate configuration.

    Clients may choose to linking an explicit application service / access mode to further suppress a menu item. For example, imagine your implementation creates a special BPA script to add a To Do Entry and would like users to use the special BPA rather than the base supplied Add dialogue for To Do Entry. The base delivered menu item for adding a To Do Entry uses the To Do Entry maintenance object application service with the Add access. Users need that security setting to be able to add a To Do Entry even from a different dialogue. To suppress the base Add dialogue, override the setting for the base supplied menu item for To Do Entry Add with a special application service and access mode. Then define a menu entry for the new special BPA for adding.
  - Zones define an application service.


Copyright © 2026, Oracle and/or its affiliates.

        * For zones linked to a portal, if a user doesn't have access to the zone's application service, the zone will not be visible on the portal. In most cases the zone is delivered with the same application service as its portal. In special cases, such as the zones on the Sidebar, the product supplies separate application services for each zone allowing implementations to determine at a more granular level which users should have access to which zones.
        * For query zones that are configured on a multi-query zone, if a user doesn't have access to the zone's application service, the zone will not be visible in the dropdown on the multi-query zone. In most cases all zones in a multi-query zone define the same application service as the multi-query zone. The product may supply a special application service for one or more zones in a multi-query zone if the functionality is special to certain markets or jurisdictions and not applicable to all implementations.
        * For zones that are used by business services to perform SQL queries, the product supplies a default application service (`F1-DFLTS`). Security for these zones is not checked by the product as they are used for internal purposes.
    - For portal based pages, individual elements may be shown or hidden based on security using the **oraHasSecurity** function in a UI map's HTML or in UI Hints in a schema. Refer to Check User's Security Access for more information.
    - Business objects define an application service. If the business object defines a lifecycle, the application service must include access modes that correspond to each state. In addition, the standard maintenance object access modes of **Add**, **Change**, **Delete** and **Inquire** are included. The base product business objects are supplied with appropriate application services. In addition, implementations may override the configured application service if desired.

      > ⚠ **Caution**
      >
      > Important! If an application service supports actions that modify the database other than **Add**, **Change**, and **Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a business object that includes lifecycle states **Freeze**, **Complete** and **Cancel**). If you want to give a user access to any of these special actions, you must also give the user access to the **Change** action.

    - Batch controls define an application service which provides the ability to secure submission of individual batch processes. The application service must include an access mode of **Execute**. The base product batch controls are supplied with appropriate application services. These services will typically have an ID that matches the batch control ID.
    - Report Definition records define an application service. The application service must include an access mode of **Submit / View Report**.
- The following objects are securable but are typically executed via internal processes. The security is provided to ensure that any access to the objects from an external source is secured.
    - BPA scripts may define an application service with the access mode **Execute**. The base BPA scripts are typically not configured with any application service. An implementation may define one. Note that as mentioned above, a menu item may also be configured with an application service and access mode. This allows for a BPA that is invoked via a menu entry to be secured in more than one way.


Copyright © 2026, Oracle and/or its affiliates.

- Business Services and Service Scripts define an application service with the access mode **Execute**. This is needed for services that may be executed from an external system, for example via an inbound web service. Base business services and service scripts that are linked to an inbound web service are configured with special application service. All other business services and service scripts are delivered with a default application service (`F1-DFLTS`), which may be overridden by an implementation.
- Scripts may also check security within an Edit Data step type if there is some functionality that should or should not execute based on a user's security access. The Edit Data step would include a call to the business service `F1-CheckApplicationSecurity`
- Service Programs define an application service. As mentioned above, for Portals and Maintenance Objects, their application service is taken from the related service program. In base, specific application services are released for each of these types of service programs. All other service programs are typically delivered with a default application service (`F1-DFLTS`), which may be overridden by an implementation. Note that for service programs linked to a Business Service, the application service on the business service takes precedence when invoking the business service.

### Determining What an Application Service Secures

Because the product wants to provide as much granularity as possible so that implementations can have the greatest control over what users can do, there are a large number of application services provided out of the box and it can be daunting to understand who needs to have access to what application services to do their job. The product provides several online tools to help security administrators to understand what application services exist and what they are for.

- On the application service page itself, there is a Secured Objects zone that indicates all the metadata that is related to this application service. There are a small number of application services that are securing functionality that is not discoverable by the Secured Objects zone. For those services, the detailed description of the application service should indicate what it's for.
- A large percentage of security is related to a specific page in the system that is accessible via the menu. Because of the different layers of security, to be able to view a record you may need access to several application services: the one associated with the query portal, the one associated with the maintenance portal, the one associated with the maintenance object of the record and the one associated with the business object governing the record. When viewing a menu, there is an Application Security tab, which allows you to review each menu line and see a list of all the application services a user may need access to for viewing or maintaining data on that related page.
- The Application Security Query portal provides various filters for viewing what objects are being secured by what services. Besides menu items, it shows batch controls, reports, web services and other functionality. In addition, it allows you to view what users and what user groups currently have access to what application services and allows you to compare the security of two users.
- The User Group portal includes several zones that help security administrators to configure security options or review security configuration. The main tab includes a zone to Add Application Services. This zone only shows application services that are not already linked and provides several ways to narrow down the list of application services. The Service Manager tab has a similar zone, but breaks down the information to a combination of application service and access mode. Finally, the Secured Components tab is similar to the Application Security Query. It allows you to query the objects secured by application services in this user group.


Copyright © 2026, Oracle and/or its affiliates.

### Application Services Introduced in a Release

When new functionality is provided that includes any object that is associated with an application service, new application services may be introduced. The product release notes indicate the new application services included in the release. In addition, the product includes a special Entity Tag that includes a list of the new application services. The tag's description uses the cloud release reference rather than the on-premise release reference to identify itself.

## Field Level Security

Sometimes transaction and action security is not sufficient. There are situations where you may need to restrict access based on the values of data. For example, in Oracle Utilities Customer Care and Billing you might want to prevent certain users from completing a bill for more than $10,000. This is referred to as "field level security".

Field level security can be complex and idiosyncratic. Implementing field level security always requires some programming by your implementation group. This programming involves the introduction of the specific field-level logic into the respective application service(s). The recommendation is to use a business object validation algorithm for checking field level security.

> **Note**
>
> The check for field level security must exist in validation logic to ensure that data is not added or updated if it does not pass the field level security rules.
>
> It is not straight forward to implement user interface changes related to field level security, such as protecting elements under certain conditions. This is because overriding the base product delivered user interface to check for implementation specific conditions is not always supported, depending on how the user interface is built, or requires a full duplication of product artifacts, which is not recommended.

The definition of a user's access rights is performed using the same transactions used to define transaction / action level security. This is achieved as follows:

- Create a security type for each type of field-level security.
- Define the various access levels for each security type. For example, assume you have some users who can complete bills for less than $300, and other users who can complete bills for less than $1,000, and still other users who can complete bills for any value. In this scenario, you'd need 3 access levels on this security type:
  - Level 1 (lowest): May authorize bills <= $300
  - Level 2 (medium): May authorize bills <= $1,000
  - Level 3 (highest): May authorize all bills
- Link this security type to each application service where this type of field level security is implemented. This linkage is performed on the security type transaction.
- Defining each user group's access level for each security type (this is done for each application service on which the security type is applicable).


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Highest value grants highest security. The system expects the highest authorization level value to represent highest security level. Moreover, authorization level is an alphanumeric field so care should be taken to ensure that it's set up correctly.

## Importing Security Configuration from an External Source

The product provides support for importing security information from an external source. In all cases, the product uses a special user BO `F1-IDMUser` to create the user and the recommendation is to take advantage of the Template Userfunctionality. More information about the out of the box support for using this feature is described in the related links.

- If your organization uses Lightweight Directory Access Protocol (LDAP), you can import your existing LDAP users and groups into the system. Once imported, all user and group functions are available. You can import a user group, or a single user. You can resynchronize your LDAP users and groups at any time.

  > **NOT_SUPPORTED**
  >
  > For more information refer to LDAP Integration.

- The system provides an integration with Oracle Identity Manager. When a user is created in the identity manager product, its information can automatically be interfaced to the product. Once the user is successfully created in the system, all functions are available.

  > **NOT_SUPPORTED**
  >
  > For more information refer to Oracle Identity Manager Integration.

- The system provides a web service with add, update and get operations that follow the System for Cross-domain Identity Management (SCIM) specification for provisioning users.

  > **NOT_SUPPORTED**
  >
  > For more information refer to Provision Users with SCIM Open Standard.

# Defining Application Services

Please refer to Understanding Application Security for a description of how application services are used when you grant user groups access rights to transactions and the significance of an application service's access modes.

You can access the portal using **Admin**, and then **Security**, and then **Application Service.** You are brought to a query portal with options for searching for a specific application service. Once a record has been selected you are brought to the maintenance portal to view and maintain the selected record.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- The **Application Service** zone provides the information about the application service and its access modes.
- The **Secured Objects** zone displays the object (or objects) that are secured by this application service. Refer to Secured Objects and Application Services for details about the types of objects that may be secured.

The **Application Security** tab includes zones to view and manage which user groups have access to the application service and which ones do not have access .The following zone may appear as part of this tab page:

- The **User Groups Linked** zone lists the user groups that currently have a link to the application service. Note that expired links are included when the zone is first displayed. Use the zone filters to exclude expired links and to limit the user groups shown by user ID, user group ID, and user group description.
- The **User Groups not Linked** zone lists the user groups that do not have a link to the application service. Use the zone filters to limit the user groups shown by user ID, user group ID, and user group description.

# Defining Security Types

Security types are used to define the types of field level security.

> **Note**
>
> **Programming is required.** You cannot have field level security without introducing programming code. Refer to Field Level Security for more information on how security types are used to define field level security.

- Security Type - Main

## Security Type - Main

Select **Admin**, and then **Security**, and then **Security Type** to define your security types.

**Description of Page**

Enter a unique **Security Type** and **Description**.

Use the **Authorization Level** grid to define the different authorization levels recognized for this security type. Enter an **Authorization Level Number** and its **Description**.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Programming is required.** Note that the values that you enter are not interpreted by the system itself, but by the user exit code used to implement the special security. Check with the developer of the user exit logic for the correct values. Refer to Field Level Security for more information on how security types are used to define field level security.

Use the **Application Services** grid to define the application service(s) to which this security type is applicable. If this application service is already associated with user groups, you must update each user group to define their respective security level. This is performed on the User Group portal.

# Defining User Groups

A user group is a group of users who have the same degree of security access. Think of a user group as a "role"; associated with a role are:

- The users who play this role
- The application services to which the role's users have access (along with the actions they can execute for each service and their field level security authorization levels).

Refer to Secured Objects and Application Services and Field Level Security for more information about access modes and security types.

> **Caution**
>
> Important! If an application service supports actions that modify the database other than **Add**, **Change**, and **Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a transaction that supports actions in addition to **Add**, **Change**, and **Inquire** (e.g., **Freeze**, **Complete**, **Cancel**). If you want to give a user access to any of these additional actions, you must also give the user access to the **Inquire** and **Change** actions.

This portal allows a security administrator to view and manage information about a user group.

You can access the portal from the **Admin Menu**, and then **Security**, and then **User Group**. You are brought to a query portal with options for searching for a specific user group. Once a user group has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Caution**
>
> Application services may not be changed or removed from any Base Delivered User / User Groups.

The following zones may appear as part of the portal's **Main** tab page

- **User Group.** This zone provides information about the selected user group.


Copyright © 2026, Oracle and/or its affiliates.

- **Application Services.** This zone lists application services linked to the selected user group. You may use this zone to edit the information for an specific application service or use mass actions to remove selected services or update their effective date.
- **Add Application Services.** You may use this zone to add application services to the selected user group.

The following zones may appear as part of the portal's **Users** tab page

- **Users.** This zone lists users linked to the selected user group. You may use mass actions to remove selected users or update their effective date.
- **Add Users.** You may use this zone to add users to the selected user group.

The following zones may appear as part of the portal's **Service Manager** tab page

- **Application Service / Access Modes.** This zone lists application service and access mode combinations currently linked to the selected user group. You may use mass action to remove a selected set of records.
- **Add Application Service / Access Modes.** You may use this zone to add application service access mode records.
- **Security Type Authorization.** This zone maintains field level security for application services included in the selected user group.

The following zone may appear as part of the portal's **Secured Components** tab page

- **Application Components.** This zone allows you to review various types of application components secured by the selected user group. The query is similar to the Application Security Query restricted to the current user group.

# Defining Users

The user maintenance transaction is used to define a user's user groups, data access roles, portal preferences, default values, and To Do roles. To access the user maintenance transaction, select **Admin**, and then **Security**, and then **User**.

The user maintenance transaction is the same transaction invoked when the user launches Preferences from the user menu or shortcut key.

# Application Security Query

This portal allows a security administrator to review application security configurations across users, user groups and the various types of application components they secure.

While application security is configured at the granular level of an application service and its access modes, it is more intuitive to review access rights to top level broader components:

- **Menu line.** Several application services may be related to a single menu line and impact the ability to properly leverage various aspects of its functionality:
  - The application service of the transaction page or portal, which also controls the visibility of the menu line or its 'search' part when the line is composed of 'add' and 'search' items.
  - The application services associated with any zones on that portal.
  - The application service of the underlying maintenance object when applicable, which also controls the 'add' item of the menu line.
  - The application services of business objects for the underlying maintenance object.


Copyright © 2026, Oracle and/or its affiliates.

  - The application services of portals used to maintain these business objects.
  - The application services of related portals.
  - The application services of related maintenance objects, their business objects and portals.
  - and more

  You may review the list of application services associated with a menu line on the **Application Security** tab on the **Menu** portal.

- **Sidebar Zone.** This is a regular or context sensitive zone that the user has access to on the sidebar area.
- **System.** This is a Toolbar or other system wide option a user is allowed to use.
- **Batch Control.** This is a batch process a user is allowed to submit.
- **Report Definition.** This is a report a user is allowed to submit.
- **Web Service.** This is a web service a user is allowed to invoke.
- **Miscellaneous.** This is a special service that a user is allowed to invoke.
- **Internal.** This is an internal item that is secured by default but is assumed to be called in a context that does not require security check.
- **Unclassified.** Some application services may not be related to any type of application component and as such remain unclassified. For example, there are situations where previously released services are enhanced to be associated with dedicated application service in a later release. In this situation the services may retain their original or even custom application service and the new application services may appear as not associated with any component.

Refer to Secured Objects and Application Services for details about the types of objects that may be secured.

Open this page using **Admin**, and then **Security**, and then **Application Security Query**.

# Row Security

Some products allow you to limit a user's access to specific rows. For example, in Oracle Utilities Customer Care and Billing, row level security prevents users without appropriate rights from accessing specific accounts.

A combination of framework configuration and configuration in your edge product is required for row level security. The following points describe the configuration:

- For each record that should be secured, associate it with an **Access Group**. Note that if your edge product supports row level security, that product is providing a link between the secure-able record and Access Group. Your access groups may be granular and only referenced by one secured record or they may be more broad and be referenced by multiple secured records that require the same type of security restriction.
- To define which users have access to the secured records, you define a **Data Access Role**. For each data access role, define which Access Groups the role has security clearance for. An access group may be linked to one or more data access roles. In addition, define the **Users** that have access rights to these secured records. When you grant a data access role rights to an access group, you are giving all users in the data access role rights to all secured records in all the referenced access groups. A user may belong to many data access roles.


Copyright © 2026, Oracle and/or its affiliates.

If your edge product supports row level security, it will include logic in the appropriate areas of the system to limit the secured rows that a user may view or maintain based on this configuration. For example, in Oracle Utilities Customer Care and Billing, throughout the system users are only able to view and maintain information about an account and any of its detail if the user is in a Data Access Role for the account's Access Group (or the account is not linked to an Access Group).

**NOT_SUPPORTED**

Refer to your product's documentation for more information on row level security, if applicable.

- Defining Access Groups
- Defining Data Access Roles

# Defining Access Groups

Refer to The Big Picture of Row Security for a description of how access groups are used to restrict access to specific objects.

Access groups control which groups of users (referred to as Data Access Roles) have rights to accounts (or other objects) associated with the access group. You can access the portal using **Admin**, and then **Security**, and then **Access Group**. You are brought to a query portal with options for searching for a specific access group. Once a record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Access Mode** zone provides the information about the access group and its data access roles.

- The **Data Access Role** collection defines the data access roles whose users have access to the access group's secured objects. Keep in mind that when you add a **Data Access Role** to an **Access Group**, you are granting all users who belong to this role access to all of the objects linked to the access group.

  **Note**

  You can also use Data Access Role - Access Group to maintain a data access role's access groups.

- The **To Do Role Assignment** collection may be used to route To Do Entries of a given To Do Type to a specific To Do Role if the objects related to the To Do Entry are associated with the access group. Refer to Adjusting the To Do Role for more information.

# Defining Data Access Roles

**NOT_SUPPORTED**

Refer to The Big Picture of Row Security for a description of how access groups are use to restrict access to specific objects.

The data access role transaction is used to define two things:


Copyright © 2026, Oracle and/or its affiliates.

- The users who belong to the data access role.
- The access groups whose accounts (or other objects) may be accessed by these users.
- Data Access Role - Main
- Data Access Role - Access Group

## Data Access Role - Main

Select **Admin**, and then **Security**, and then **Data Access Role** to define the users who belong to a data access role.

**Description of Page**

Enter a unique **Data Access Role** code and **Description** for the data access role.

The scroll area contains the **Users** who belong to this role. A user's data access roles play a part in determining the accounts (or other objects) whose data they can access.

To add additional users to this data access role, press the add button and specify the following:

- Enter the **User** ID. Keep in mind that when you add a **User** to a **Data Access Role**, you are granting this user access to all of the accounts (or other objects) linked to the data access role's access groups.
- Use **Expiration Date** to define when the user's membership in this data access role expires.

> **Note**
>
> **Also maintained on the user page.** You can also use User - Access Security to maintain a user's membership in data access roles.

## Data Access Role - Access Group

Select **Admin**, and then **Security**, and then **Data Access Role** and navigate to the **Access Groups** tab to define the access groups whose accounts (or other objects) may be accessed by the users in this data access role.

**Description of Page**

Use the **Access Group** collection to define the access groups whose objects can be accessed by this role's users. Keep in mind that when you add an **Access Group** to a **Data Access Role**, you are granting all users who belong to this role access to all of the accounts (or other objects) linked to the access groups.

> **Note**
>
> You can also use Access Group - Main to maintain an access group's data access roles.

# Data Privacy

Personally Identifiable Information (PII) is any information that identifies or could be used in combination with other information to identify an individual. Many countries have data privacy


Copyright © 2026, Oracle and/or its affiliates.

laws governing the use of such information. A key aspect of these laws is an organization's obligation to erase personal information when there is no compelling reason to retain it. This obligation can be fulfilled either by deleting data or altering the data in such a way that it is no longer possible to associate the data with the individual.

Organizations are likely to have other legal obligations that impact how long they retain personal data. For example, there could be financial audit requirements that oblige the organization to retain relevant information for a fixed number of years after termination of a contract.

Object Erasure is designed to address right to erasure issues, with a combination of configuration and processes that may be extended to implement the appropriate policies for the PII within your data.

- Encryption and Masking
- The Approach to Implementing Object Erasure
- Data Redaction for Export and Logging
- Defining Redaction Rules
- Obfuscation Configuration

# Encryption and Masking

"Encryption" refers to encrypting data stored in the database using an encryption key. There are two different types of encryption described in the sections below. System encryption refers to columns in the system identified by the product to use encryption. Application encryption refers to the ability for an implementation to configure fields and elements that should be encrypted in the database.

"Masking" refers to overwriting all or part of an un-encrypted field value with a masking character. For example, perhaps only the last 4 digits of a tax ID number are visible with the other digits changed to an asterisk. The system provides support for masking fields on the user interface that may be stored as plain text in the database. In addition, there are cases where encrypted fields are shown to the user interface using masked values rather than the encrypted value.

The following sections provide more information about each feature.

- System Encryption
- User Interface Masking
- Application Encryption

## System Encryption

The system automatically encrypts certain fields captured in various option tables or context tables. This is mainly used for passwords. For example, passwords captured in Message Sender context or password

In addition, batch control supports configuring a security option for parameters that capture sensitive information, such as a password. Refer to Defining Batch Controls for more information.

It is also possible to enable system encryption using the characteristic type `F1-PWD`. However, the maintenance object must include specific code to enable system encryption for characteristics of this type. In Oracle Utilities Application Framework, the only maintenance


Copyright © 2026, Oracle and/or its affiliates.

object that supports this is extendable lookup. Refer to Extendable Lookup Advanced Topics for more information.

## User Interface Masking

The functionality described in this section is used to take data that is stored in plain text in the database and mask the value before it is presented to a user (or an external system). This feature includes the ability to allow some users to view the data unmasked using security configuration. The system allows different masking rules to be applied to different fields. For example, a credit card number can be masked differently than a social security number.

The following topics describe how to mask field values.

### Identify the Data to be Masked

Identify the data that is stored as plain text, but should be masked for display to users. For example, imagine that you have identified that Credit Card Numbers and a person's federal ID number (for example, in the United States, the Social Security Number or SSN). Each field identified may be displayed and maintained in different user interfaces throughout the system, but the masking rules for a given field are probably uniform regardless of where the data is displayed.

**Primary keys cannot be masked.** A field defined as a unique identifier of a row cannot be configured for masking. Masking a field that is part of the primary key causes a problem when attempting to update the record. This restriction also applies to elements that are part of a "list" in an XML column on a maintenance object. One or more elements in the list must be defined as a primary identifier of the list. Be sure that primary key elements in the list are not ones that require masking.

**List members that contain different "types".** Consider a page with a list that contains a person's identification numbers. You can set up the system so that a person's social security number has different masking rules than their drivers license number. If your implementation has this type of requirement, the list of masked fields should contain an entry for each masking rule.

For each field, if there are some users that may see the data unmasked on one or more of the user interfaces, then security configuration is required. If the value of a field should be masked for all users across all pages in the application, then the security configuration is not needed.

### Security Configuration

Define a security type for each field with two authorization levels:

- 1 - Can only see the element masked
- 2 - Can only see the element unmasked

Link all of the security types to an application service of your choosing. We recommend linking every masking-oriented security type to a single application service (e.g., `CM_MASK`) as it makes granting access easier.

For each security type, identify which users can see its data unmasked and which users can only see its data masked. If the masked and unmasked users fit into existing user groups, no additional user groups are necessary. Otherwise, create new user groups for the masked and unmasked users.

After the user groups for each security type are defined, link each user group to the application service defined above. When a user group is linked to the application service, you will define the authorization level for each security type linked to the application service. If a user group's


Copyright © 2026, Oracle and/or its affiliates.

users should see the security type's field values unmasked, set the authorization level to 2; otherwise set it to 1.

> **Note**
>
> Updating masked values. The system supports allowing a user that sees a masked value to update the value. For numbers that are masked, the masked value is typically all 9s. If there is some chance that a user is allowed to update a masked number, the recommendation is that the user is configured with authorization to see the number unmasked. Otherwise, there is a corner case where the user sees the masked number and needs to update the number to a value that exactly matches the masked value. In this case, the update won't go through because the system doesn't recognize it as a change.

> **Note**
>
> Flush the cache. Remember that any time you change access rights you should flush the security cache (by entering flushAll.jsp on the URL of the application) if you want the change to take effect immediately.

### Configure a Masking Algorithm

A data masking algorithm (using the algorithm entity value of **Feature Configuration - Data Masking**) must be created for each combination of masking rules and security type. These algorithms determine if a user has the rights to view a given field unmasked, and, if not, how the field should be masked.

The base package provides two algorithm types. Both support configuration to allow a set of users to see the data unmasked. Parameters capture the application service, security type and authorization level defined above used to evaluate this.

- `F1-MASK`: this is for masking alphanumeric data. Its parameters are designed to handle most masking needs. Parameters allow you to configure how much of the data to mask and what masking character to use. Refer to the algorithm type description for more information.
- `F1-MASKNBR`: this is for masking numbers. Refer to the algorithm type description for more information.

### Determine How the Fields are Displayed

The masking configuration differs based on how a field is retrieved for access to the user interface. So for the masking of one "logical" field (like a person's SSN), there may be multiple configuration entries required to cover all the access methods. Review each user interface where a given field is displayed and create the following categories:

- The field is an element that is retrieved by invoking a business object, a business service, or a service script
- The field is stored as an ad hoc characteristic
- The field is displayed on a fixed maintenance page (and is therefore retrieved by invoking a page service)
- The field is displayed on a fixed search page (and is therefore retrieved by invoking a search service)


Copyright © 2026, Oracle and/or its affiliates.

### Create a Feature Configuration for Each Masked Element

Create a feature configuration with a Feature Type of **Data Masking**. An option entry with option type of **Field Masking** is needed for every combination of field to mask and the method used to display the data. The value will contain mnemonics that reference the appropriate data masking algorithm along with configuration that differs depending on how the field is retrieved for display as described below.

### Schema Based Object Field Masking

For data that is accessed via a schema-based object call and displayed in a UI map, the field to be masked must reference a meta-data field name in its schema definition: `field="fld_name", alg="algorithm name"`

If the element references an mdField in the schema, that is the field used to identify the masking rule. If there is no mdField reference but only a mapField reference, that is the field used to identify the masking rule. For example, if you want to mask a credit card number, let's assume that field is defined in the schema is the following:

```
<creditCard mdField="CCNBR" mapField="EXT_ACCT_ID"/>
```

In this case, the option value should be `field="CCNBR", alg="algorithm name"`. An option value of `field="EXT_ACCT_ID", alg="algorithm name"` would not result in masking.

A "where" clause may also be specified. This is useful for data that resides in a list where only data of a certain type needs to be masked: `field="fld_name", alg="algorithm name", where="fld_name='value'"`

For example, person can have a collection of IDs and only IDs of type 'SSN' (social security number) should be masked. If the person data including its collection of person IDs is displayed on a UI map via a business object call, let's assume the collection is defined in the following way:

```
<personIDs type="list" mapChild=CI_PER_ID">
  <isPrimaryId mapField="PRIM_SW"/>
  <idType mapField="ID_TYPE_CD"/>
  <personIdNumber mapField="PER_ID_NBR"/>
</personIds>
```

The option value may look like this: `field="PER_ID_NBR", alg="algorithm name", where="ID_TYPE_CD='SSN'"`

Please note the following important points for schema based masking:

- **Limitation of 'where' field** Although the main use of a 'where' clause for schema oriented elements is to mask certain elements in a list based on a 'type', it is also possible to mask a single field in the schema based on the value of another field. For example, imagine that a customer submits a registration form that defines an ID type and ID value. Although this data is not in a list, the implementation may still want to only mask the ID value if the ID type is "SSN". The framework is only able to mask an element in the schema based on a 'where' clause if the element in the 'where' clause is a "sibling" in the schema.
  - If the element to be masked is in a list, the element in the 'where' clause must be in the same list.


Copyright © 2026, Oracle and/or its affiliates.

- If an element to be masked maps to a real column in a table, the element in the 'where' clause must also map to a real column in the table.
- If an element to be masked maps to and XML column in the table as a single element, the element in the 'where' clause must map to the same XML column as a single element.

- **Multiple feature option entries for the same field.** It's possible that different schemas in the system have a similar type of data that may be masked based on different conditions. For example, imagine that an implementation has different schemas that captured or referenced person identifiers in different ways:
  - One schema captures a single person ID without any corresponding "type" record and it should always be masked using Algorithm CM_SSN_MASK:

    ```
    <personSSN mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
    ```

  - One schema captures a person ID and a corresponding ID Type and it should be masked with Algorithm CM_SSN_MASK if the type is "SSN" and masked with algorithm CM_FEIN_MASK if the type is "FEIN".

    ```
    <personIdType mapXML=BO_DATA_AREA mdField=ID_TYPE_CD/>
    <personId mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
    ```

  - One schema captures a person ID and a corresponding ID Type and it has the same masking rules as the previous schema, but a different field name is used for the ID Type code. (This scenario could happen if for example a different label is desired for ID Type on the user interface for this schema.)

    ```
    <personIdType mapXML=BO_DATA_AREA mdField=CM_ID_TYPE/>
    <personId mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
    ```

  For this scenario, the feature options may look like this:

  1. `field="PER_ID_NBR", alg="CM_SSN_MASK"`
  2. `field="PER_ID_NBR", alg="CM_SSN_MASK", where="ID_TYPE_CD='SSN'"`
  3. `field="PER_ID_NBR", alg="CM_FEIN_MASK", where="ID_TYPE_CD='FEIN'"`
  4. `field="PER_ID_NBR", alg="CM_SSN_MASK", where="CM_ID_TYPE='SSN'"`
  5. `field="PER_ID_NBR", alg="CM_FEIN_MASK", where="CM_ID_TYPE='FEIN'"`

  For each schema, the system will first find whether the element applies to any masking option. It will find 5 masking options for the field PER_ID_NBR. Then it will determine if any sibling elements match the 'where' clause.

  - If more than one sibling element matches a 'where' clause, a runtime error is issued. For example if a schema has an element that references "mdField=ID_TYPE_CD" and an element that references "mdField=CM_ID_TYPE", this is an error. Additionally, if multiple elements reference mdField=ID_TYPE_CD", this is an error.
  - If one and only one sibling element matches a 'where' clause, the value of the element is compared to the values defined in the 'where' clause. If it finds a match on the value, the appropriate masking algorithm is applied. If no match is found (for example, the Person ID Type is "LICENSE") the element is displayed as is.


Copyright © 2026, Oracle and/or its affiliates.

- If no sibling element matches a 'where' clause and a feature option exists with no 'where' clause (option 1 above), then the masking algorithm of the option with no 'where' clause is applied.

- **Changing the value in the 'where' clause.** If your implementation has some users that are allowed to change records where some data is masked based on a condition, it is recommended to design the user interface to reset the masked value when the value in the 'where' clause changes. For example, if a user is prevented from viewing a person's social security number, but the user is allowed to make updates to the person's record, changing the value of the Person ID Type should reset the Person ID Number. This would ensure that the user does not 'unmask' the social security number by simply changing the ID Type.

**Records Maintained Using Page Maintenance**

For data that is accessed via a page maintenance service call, indicate the table name and the field name where the data resides: `table="table_name", field="fld_name", alg="algorithm name"`

For example if the Person record and its collection of identifiers are displayed and maintained using page maintenance, the option value should be `table="CI_PER_ID", field="PER_ID_NBR", alg="algorithm name"`

A "where" clause may also be specified: `table="table_name", field="fld_name", where="fld_name='value'", alg="algorithm name"`

This is useful for data that resides in a child table where only data of a certain type needs to be masked. For the person ID example, `table="CI_PER_ID", field="PER_ID_NBR", alg="algorithm name", where="ID_TYPE_CD='SSN'"`

**Characteristic Data**

For data that is stored as a characteristic, simply indicate the characteristic type: **CHAR_TYPE_CD='char type', alg="algorithm name"**

This needs to be defined only once regardless of which characteristic entity the char type may reside on. Note that only Ad hoc characteristics are supported.

**Masking Fields in Explorer Zones or Info Strings**

In explorer zones data is often retrieved using SQL directly from the database. No masking is applied automatically in this case. If there is data in the explorer zone results that should be masked, the masking must be applied by calling a business service.

Similarly, an MO Info algorithm may not use BO interaction to get data. It may access data using SQL for efficiency purposes. No masking in applied when retrieving data via SQL. To apply masking to a string prior to including it in an info string, the masking must be applied by calling a business service.

The system supplies two business services to be called to determine if masking rules apply for a specific field.

- `F1-TableFieldMask`. Mask a Table field. This business service receives a table name, field name and one or more field values. If masking applies it returns the masked value.
- `F1-SchemaFieldMask`. Mask a Schema field. This business service receives a schema name and type, XPath and field value. If masking applies it returns the masked value.

**Search Service Results**

For data that is displayed on a 'fixed' search page, it is retrieved via a search service call. Indicate the search name and the appropriate field to mask along with the masking algorithm.


Copyright © 2026, Oracle and/or its affiliates.

For example: `search="SearchServiceName", field="PER_ID_NBR", where="ID_TYPE_CD='SSN'", alg="algorithm name"`

To find the name of the search service, launch the search in question, right click in the filter area and choose **Inspect**. Search for "serviceName". The service name is listed there. To find the field name to mask, search for "Widget Info". There should be two results found, one for the filter area and one for the results area. This results area has the text "SEARCH_RESULTS" as prefix for each field. The field names are the ones after the x$. Do not reference the x$ when defining the field name. Note, the "where" statement can only apply to fields that are also part of the search results.

#### Additional Masking Information

The following points provide additional information to assist in your masking configuration:

- If the demonstration database includes a **Data Masking** feature configuration, review the settings because it will probably contain masking rules that will match your own.
- On data input pages, a user might be able to enter or change masked data, such as a bank account number, but not be able to subsequently see what they added or changed.
- External systems can request information by performing a service call via a web service. Please keep in mind that some web service requests require data to be masked and some do not. For example, a request from an external system to synchronize person information needs the person's social security number unmasked; whereas a request from a web self service application to retrieve the same person information for display purposes needs the person's social security number masked. To implement this type of requirement, different users must be associated with each of the requests and these users must belong to separate user groups with different access rights.
- If a maintenance object (MO) contains a field that holds an XML document and a service call invokes the MO's service program directly, the system will mask individual XML elements in the field if a **Determine BO** algorithm has been plugged into the maintenance object and the element(s) in the respective BO schema have been secured as described above.

## Application Encryption

The functionality described in this section allows implementations to configure fields to encrypt when storing it in the database. This functionality is mutually exclusive from the User Interface Masking functionality. This feature also supports encrypting specific elements stored within a CLOB or XML column.

The following points highlight the features of the encryption functionality:

- The encryption key is defined using a key ring. Refer to Field Encryption for information about defining encryption key rings.
- When a field is configured to be encrypted, the encrypted data is stored in a special encryption field that is not the source field (the one exposed to the user on the user interface). The source field captures the data as masked. Because a special field is required to support encryption, the product must provide support for that field to be encrypted.
- For encrypted data that must allow searching, the system supports capturing a hash value in a special field. The product must provide support for this functionality. Besides providing a special field to capture the hash value, base search functionality for that data must also cater for this configuration.


Copyright © 2026, Oracle and/or its affiliates.

- The system supports encrypting data that is captured as an element within an XML field. If the XML field is provided in a schema owned by the product, then the product must provide specific support for the capture of the encrypted data.
- For backward compatibility, the system also supports encryption keys and hash keys defined using a keystore. This is not the recommended method going forward, but further information about defining the keystore in the system can be found in the Installation Guide. Refer to Moving from Keystore to Key Ring for information about adopting the key ring functionality if you are currently using the keystore.

The following sections provide additional information about the support for encryption provided by the framework. Refer to the security chapter of the administration guide for your particular product for more information.

### Encrypting and Masking the Data

When a product enables encryption for a given field, a special encryption field should be created to capture the encrypted value. Because encrypting is optional, the source field (the one exposed to the user) should not be this special encrypted field. If encryption is configured, the system will internally populate the encrypted field. The source field will be populated with asterisks by default. That way the masked data is what is shown to the user on the page rather than the encrypted value.

The following points highlight how the system behaves when encryption is configured and when it is not. Assume as an example, the field is a credit card number. The user views and populates a field with the field name CC_NBR. The table also has a second field ENCR_CC_NBR. A user populates the credit card number:

- If encryption is not configured, CC_NBR will be updated with the entered credit card number and ENCR_CC_NBR will be empty. Note that in this case, an implementation may choose to configure user interface masking.
- If encryption is configured, CC_NBR will be updated with '**********' and ENCR_CC_NBR will contain the encrypted value. The asterisks for the standard field will fill the full field size up to 50 characters.

If for some reason the standard masking using all asterisks is not desired, the system supports supplying an explicit masking algorithm using the same Feature Configuration - Data Masking plug-in spot used for User Interface Masking.

> **Warning**
>
> Unlike user interface masking, the masking of encrypted fields is not driven by security. The data stored in the source field for all encrypted data should be masked. Be sure not to configure security authorization logic in algorithms used for this type of masking.

### Feature Option Configuration

Create a feature configuration with a Feature Type of **Encryption**. For each source field you are encrypting, enter an option with option type of **Field Encryption**. The value will contain mnemonics that reference the appropriate encryption key ring along with configuration related to the field and its table location. Unlike the user interface data masking, the configuration for data encryption is related to how the data is stored rather than how it is displayed. In addition, each entry may define an explicit masking algorithm to override the default and if supported, may also define a hash field and hash key ring.

Refer to Field Encryption for information about defining encryption key rings.


Copyright © 2026, Oracle and/or its affiliates.

For data that is stored in a specific column on a table, an explicit field to capture the encrypted value must exist. Indicate the table name, source field name and encrypted field name along with the key ring: **table='table_name', field='fld_name', encryptedField='encr_fld_name', keyRing='key ring code'**

A "where" clause may also be specified when data resides in a child table and only data of a certain type needs to be encrypted.

Example, **table='CI_PER_ID', field='PER_ID_NBR', encryptedField='ENCR_PER_ID_NBR', keyRing='CM-SymmetricKey', where='ID_TYPE_CD='SSN''**

For data that is stored in an XML column in a record, the source field to be encrypted must reference a meta-data field name in its schema definition along with the element that captures the encrypted data and the alias: **field='field_name', encryptedField='encr_field_name', keyRing='key ring code'**

The syntax for adding a reference to a masking algorithm is **maskAlg='algorithm name'** .

The syntax for adding configuration for capturing a hash value for searching purposes is **hashKeyRing='key ring code' hashField='HASH_FLD_NAME'**.

The following is an example of configuration that uses all the possible options (specific masking algorithm, where clause and hash field support):

**table='CI_PER_ID', field='PER_ID_NBR', keyRing='CM-SymmetricKey', encryptedField='ENCR_PER_ID_NBR', hashKeyRing='CM-HashKey' hashField='HASH_PER_ID_NBR', where='ID_TYPE_CD=SSN', maskAlg='CM-PERIDMASK'**
For backward compatiblity, the product supports defining an encryption key using a keystore. For implementations that continue to use this configuration, you reference the alias or hash alias instead of the key ring. The following is an example of the configuration that uses all the possible options, but uses the alias instead of a key ring.

**table='CI_PER_ID', field='PER_ID_NBR', alias='aliasKey', encryptedField='ENCR_PER_ID_NBR', hashAlias='hashAliasKey' hashField='HASH_PER_ID_NBR', where='ID_TYPE_CD=SSN', maskAlg='CM-PERIDMASK'**
In addition, when using an alias, the encrypted field may optionally be "wrapped" with a special marker (e.g., ENC{} ) to indicate that the field value is encrypted. This can be specified by the setting wrap="true" or wrap="false". The default is false. This should be set to false unless additional processing in your code is included to handle the special marker.

Refer to Moving from Keystore to Key Ring for information about adopting the key ring functionality if you are currently using the keystore.

### Searching by an Encrypted Value

If the product supports a hashed value for an encrypted field for searching purposes, the following points highlight explorer zone configuration for this purpose

- The user filter value should reference the source field and should include an additional `encrypt=` mnemonic. For example

  ```
  type=STRING
  label=PER_ID_NBR
  encrypt=[CI_PER_ID,PER_ID_NBR,ID_TYPE_CD,F1]
  ```

  Refer to User Filters for more information.
- The SQL should include the hashed value in the WHERE clause. Note that because encryption is optional, a product zone that includes searching by a field eligible for


Copyright © 2026, Oracle and/or its affiliates.

encryption will include finding a match for the filter in the source field (as plain text) or in the hashed field. For example:

```
WHERE
   [(F2) (ID.PER_ID_NBR =:F2 OR ID.HASH_PER_ID_NBR = :F2)]
```

### Customizing Encryption Algorithm

This feature is only applicable to implementations that use the keystore. This is not the recommended configuration going forward. The text is kept here for information purposes.

Although the encryption algorithm to use with a given key can be gleaned from the key in the keystore, there is sometimes extra information associated with an algorithm that might need to be used to encrypt or decrypt data.

The system provides a feature configuration option for the **Encryption** feature type using the option type **Algorithm Info** that can be used to adjust the behavior of the encryption.

- You can modify the default mode and padding of the encryption algorithm.
- If a key will be used to digitally sign anything, the signing algorithm can also be specified for the key.

For details about the syntax, refer to the feature option type's detailed description.

- Managing Encrypted Data

## Managing Encrypted Data

The sections in this topic provide information about encrypting data when first implementing the application along with various use cases that may require re-encrypting data, such as key rotation.

The batch job `F1-FLENC` (Field Encryption for a Maintenance Object) is provided to handle encryption / re-encryption of all data for a given Maintenance Object. Review your encryption configuration and identify all the maitnenance objects that should have data encrypted. Run the batch once for each maintenance object. The below sections highlight various use cases that warrant running this batch for each maintenance object.

### Initially Encrypting Data

Once your encryption / hash configuration is complete, any data added or updated from that point on is encrypted / hashed per the configuration. However, for all existing records that have data that should be encrypted / hashed, you need to run the batch job `F1-FLENC` for each maintenance object to go through and encrypt / hash the data in the existing rows.

### Mass Key Rotation

If you introduce a new encryption key or hash key, as described in Key Rotation, or you introduced a new key ring as described in Key Ring Rotation, records added or updated from that point on will be re-encrypted / re-hashed with the latest key for the respective key ring. However, to rotate the key for all existing records that are encrypted / hashed with the previous key, you need to run the batch job `F1-FLENC` for each maintenance object to go through and re-encrypt or re-hash the data in the existing rows.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> You can independently choose to introduce a new encryption key or a new hash key. The batch program will check each encrypted field or hashed field to confirm that they use the current active key and if not, the data is adjusted.

> **Note**
>
> When rotating a Hash Key, from the time the new key is activated until the time when all records have been successfully updated to use the new hash key, if any end users attempt to search for records via a hashed field, there may be inconsistent results. You should schedule rotation of hash keys to be during a time where end users are not impacted. Ideally you activate the new key and run the key rotation batch control for all maintenance objects during off hours.

**Moving from Keystore to Key Ring**

Using the key ring for defining your encryption and hash keys allows for key rotation. If your implementation has been using a keystore for field encryption / hashing and would like to use the key ring instead, you can do this as described in the following steps.

- Define a key ring for the Symmetric Key and a key ring for the Hash Key, if applicable.
- Generate and activate a key for each key ring.
- Go to the Feature Option Configuration. For each entry, add references to keyRing / hashKeyRing to define the newly created key rings. Be sure to keep the configuration for the alias mnemonic in place. The system needs this information to know how to unencrypt the existing data if it was encrypted using the keystore. Note that the system doesn't need the hashAlias if you are moving from using the keystore for the hash key to using the key ring. The system simply applies the new hash key.
- Records added or updated from that point on will be re-encrypted / re-hashed using the indicated key ring and its active key. To rotate the key for all existing records that are encrypted / hashed with the keystore alias, you need to run the batch job `F1-FLENC` for each maintenance object to go through and re-encrypt or re-hash the data in the existing rows.
- Once you have confirmed that all the data is using the key ring for encryption and hashing, you can update the feature option configuration to remove the reference to the old alias keys.

The "notes" listed in the Mass Key Rotation section also apply to this use case.

# The Approach to Implementing Object Erasure

This section describes the product approach to implementing object erasure for its maintenance objects (MOs).

> **Note**
>
> The approach outlined is intended for managing erasure of master data objects. The assumption is that Information Lifecycle Management will be used for archiving any related transactional data.


Copyright © 2026, Oracle and/or its affiliates.

Various events in the system may signal the need to erase data at some future date. For example, closing an account may indicate the need to erase both the account and person data once there is no longer a need to maintain related financial details. There may also be events that re-establish the business relationship with a person and signal that data marked for erasure should now be retained.

The Object Erasure Schedule is used to capture key details of objects with data that needs to be erased. Records in the schedule are monitored periodically to determine if the date for erasure has been reached and if so, execute the erasure logic. The system will create and maintain records in the schedule when specific events occur for an MO that is marked for erasure.

The system provides the ability to mark an MO as eligible for erasure via a number of MO configuration entries. Included in these is an option that defines the business object for the MO's erasure schedule records. In addition to defining the erasure record elements, this BO defines the monitoring process and the specific erasure logic for the MO.

The following sections provide more information about configuring and managing object erasure.

- Configuring a Maintenance Object for Erasure
- Manage Erasure Schedule Algorithm
- Monitoring the Schedule and Performing Erasure
- Erasing User Information By Obfuscation
- Viewing an Object’s Erasure Status
- Viewing Erasure Configuration
- Archiving the Object Erasure Schedule

## Configuring a Maintenance Object for Erasure

There are three types of maintenance object configuration entries that define how to manage erasure:

- An option to define the business object for erasure schedule records.
- An option to define the period between the date on which the need for erasure is detected and the date on which erasure should occur. The common logic provided by the system to add or update erasure records uses this option to determine the erasure date.
- One or more entries in the MO algorithm collection for the algorithms that determine whether a record should be scheduled for erasure, according to the current state of the record, and create or update the erasure schedule entries.

Some MOs that would commonly be regarded as eligible for erasure will be installed with a base Erasure BO already configured. The expectation is that the corresponding Erasure Period option and the algorithm for the Manage Erasure Schedule plug-in spot will be set up by your implementation with the appropriate entries for your organization's business rules.

> **Note**
>
> The presence of an Erasure Period option is the signal to the system that your implementation has enabled an object for erasure.


Copyright © 2026, Oracle and/or its affiliates.

It may not be necessary for all master data objects that contain personally identifiable information to manage their own erasure schedule. For example, the event which triggers erasure for a person record may also trigger the creation of erasure schedule entries for the person's other master data records. In this case, only the person maintenance object will need to be configured with a Manage Erasure algorithm (or algorithms). Refer to your product specific documentation for more information on the recommended approaches for your product's master data objects.

The portal Maintenance Object Configuration includes a zone that shows all the maintenance objects eligible for object erasure. You may use this zone to review or change the configuration for any of the maintenance objects.

## Manage Erasure Schedule Algorithm

Algorithms of this type are responsible for determining a record's erasure status and creating or maintaining an entry in the Object Erasure Schedule for the record. They are triggered when certain events occur in the system for maintenance objects that are eligible for erasure. Multiple algorithms can be configured for the plug-in spot.

Depending on the determination made by the algorithm (or algorithms), a number of different actions may need to be taken. For example:

- A new schedule record may need to be added
- An existing schedule record may need to be updated with a new erasure date
- A pending record may need to be discarded if erasure is no longer applicable
- A discarded record may need to be reactivated if erasure now applies again

The base business service `F1-ManageErasureSchedule` performs the common logic to handle the possible actions and is recommended for use by these types of algorithms. The service can also optionally create a log entry for information purposes. Refer to the description of the business service for full details.

In certain circumstances, if one record is scheduled for erasure, other related records may need to be evaluated. The base business service `F1-ManageErasureScheduleDriver` performs the logic to execute an MO's Manage Erasure Schedule algorithms. It can be used to manage any related records from within another algorithm.

This type of algorithm is plugged into the Maintenance Object — Algorithm collection.

Your product may supply algorithms to manage the schedule for base MOs that are regarded as eligible for erasure. If your organization has special business rules that are evaluated to determine the erasure schedule for an MO, a custom algorithm can be created and applied by the implementation team.

## Monitoring the Schedule and Performing Erasure

The erasure schedule is managed using the lifecycle of the erasure record's business object. A deferred monitor process is used to periodically check for records that are due for erasure and transition them from the 'pending erasure' state to the 'erased' state. An enter plug in on the 'erased' state is responsible for performing the erasure logic.

The system provides a 'root' business object for the object erasure schedule (`F1-ErasureScheduleRoot`) which defines the lifecycle that erasure schedule business objects should follow. The system also provides a monitor process (`F1-OESMN`) which is configured on the 'pending erasure' state of the 'root' business object. This process is configured to monitor object erasure schedule records whose erasure date is on or before the batch business date.


Copyright © 2026, Oracle and/or its affiliates.

Maintenance objects that are eligible for erasure should be configured with an erasure schedule business object that uses the ‘root’ business object as its parent. The algorithm that performs the erasure processing applicable to that maintenance object must be plugged in on the ‘erased’ state of the ‘child’ business object.

The system provides an erasure schedule business object (`F1-ErasureScheduleCommon`) which is designed to erase an object by deleting the main record and any child records. Your product may supply additional erasure schedule business objects for certain use cases. Refer to your product specific documentation for more information.

## Erasing User Information By Obfuscation

Some implementations consider the base User object to have personally identifiable information. Erasing user information by deleting the records is not advisable as it can cause referential integrity problems. The recommended approach is to obfuscate the data instead.

The system provides a user erasure business object (`F1-ErasureScheduleUser`) which is designed to support obfuscating a user’s identifiable information by removing it or replacing it with a non-identifiable value if the field is required.

If the User maintenance object is configured to be eligible for erasure in your implementation, the manage erasure schedule algorithm for the record will be invoked whenever the **Enable** flag is changed for a record on the User maintenance page. The system provides a base algorithm (`F1-OBJERSUSR`) specifically for the User MO. This algorithm will add an entry to the object erasure schedule if the user record is disabled or deactivate an existing erasure schedule record if a disabled user record is enabled again.

## Viewing an Object’s Erasure Status

The system provides a portal for viewing and editing an erasure schedule record. The portal is accessed from the **Object Erasure Schedule** sidebar zone. This is a context sensitive zone which only appears when accessing a record for a maintenance object that is configured to be eligible for erasure. It displays the information for the erasure schedule entry for the record in context, if applicable.

To add the zone to an eligible maintenance object:

- Navigate to Context Sensitive Zone and search for the navigation key for the maintenance object.
- Add the Object Erasure Schedule zone `F1-OBJESRSCD` to that navigation key.

## Viewing Erasure Configuration

The erasure configuration may be viewed in the Maintenance Object Configuration portal.

## Archiving the Object Erasure Schedule

The system provides the ability to archive older records in the object erasure schedule using Information Lifecycle Management. The object erasure schedule maintenance object is configured with a base ILM eligibility algorithm and an ILM crawler batch control.

The base object erasure schedule BO is configured to set the ILM archive switch when an erasure schedule record enters a final state and to reset it if the record becomes active again.

Refer to The Approach to Implementing Information Lifecycle Management for more information on archiving functionality.


Copyright © 2026, Oracle and/or its affiliates.

# Data Redaction for Export and Logging

Personal identifiable information (PII) requires special handling when exported out of the application and written to files, including log files. Information that is encrypted at the database level remain encrypted and therefore protected when exported. With respect to non-encrypted information, the need to obfuscate it for added protection depends on how the exported data is intended to be used. If the exported information is part of an integration flow with an external system then the data should remain unchanged. However, when exporting data from a production environment for testing purposes sensitive information should be anonymized before they can be imported to a lower environment or written to logs for troubleshooting purposes.

The following are key aspects of this functionality:

- Redaction rules define how specific fields should be redacted when needed.
- When redaction rules exist, data redaction is performed by default by specific tools for export and logging. In absence of redaction rules, data is exported in its original form.
- Redaction rules are customer specific and may only be relevant to environments that contain sensitive data. As such, they are not released with the base product. Review your data redaction requirements and configure appropriate rules in relevant environments.
- Certain users may be allowed to export data in a non-redacted form based on application security.

The following sections provide additional information about this topic.

### Redaction Rules Are Customer Specific

Redaction rules identify which fields should be redacted and how they should be anonymized. The list of fields to redact and the function to use to manipulate their values are customer specific and therefore are not released with the base product. A sample set of rules are provided as accelerator data for reference only.

It is important that you review your data redaction requirements and configure rules that reflect your business needs.

### Redaction By Default

Processes that support data redaction on export and when logging do so by default when redaction rules exists. When rules are not defined, data is exported as is. Note that for this reason it is unnecessary to define such rules in lower environments that do not contain sensitive information.

### Specific Export Tools Only

Data redaction is supported by the following tools only:

- **Content Migration Assistant (CMA).** Depending on application security, a user may indicate on a specific data set export request that data should be exported in its original form.
- **General Data Export (GDE) batch processes.** A batch parameter may be used to explicitly request that data is exported in its original form. This option is also protected by application security and enforced at batch submission time.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This type of process establishes a single export stream that may be sent to multiple targets. Review these integrations to ensure they do not have conflicting data redaction requirements and set the batch parameter accordingly.

- **Debug logging**. By default, data is redacted base on redaction rules when writing details to the various logs.
  - In an environment that is not a live production environment, a user with appropriate security may use a feature configuration to turn this off. Find the Feature Configuration record for the **General System Configuration** feature type. (It may need to be defined if it does not exist). Choose the option type **Turn Off Log Redaction** and define a value of **Y**.
  - The system uses a property (isLiveProduction) to check if an environment is a live production environment. In a cloud implementation, the development operations team sets the property when your implementation has indicated that you are live. In an on premises implementation, it is recommended to set this property when you are live. Refer to the system administration guide for more information.

### Different Types of Redaction Rules

Sensitive information may reside in various types of storage fields which impact the way they are identified for redaction. You may define rules that target fields on specific tables or provide a more global rules that applies to all references of a specific field.

The application supports the following types of redaction rules:

- **Physical Field.** This type of rule applies to a value residing in a field on a specific table. An optional filter expression may be provided when only records of a certain type needs to be redacted.
- **XML Storage.** This type of rule applies a value associated with the specified field name that resides within an XML storage field on a specific table. This option also supports a filter expression.
- **Reference Field.** This type of rule applies to a value residing anywhere and associated with the specified field name. It does not include fields referenced in XML storage fields.

### Redaction Function

A redaction function defines the format by which a value is anonymized. The type of function used depends on the field's data type. For example, to anonymize a numeric field the redaction function should only use numeric values etc.

Refer to the **Redaction Function** extendable lookup for a list of available functions. The base product provides a sample of common functions but the list may be extended by your organization as needed.

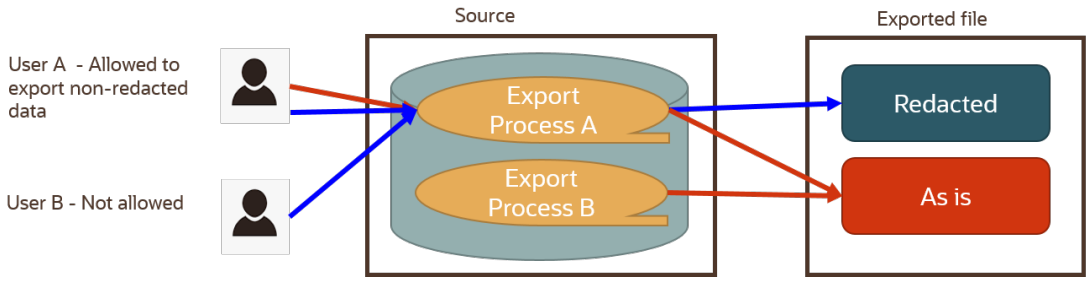
### Fields May Contain Sensitive Information

The field definition record includes an indication as to whether the field may contained sensitive information or not. Note that the indication does not imply that the field is actually used to capture such data but rather that it can potentially be used this way. Use this indication to review fields that may need to be associated with a redaction rule.


Copyright © 2026, Oracle and/or its affiliates.

### Exporting Non-Redacted Data Requires Security Rights

By default, the export tools redact data based on existing redaction rules. There are situations where certain users may be allowed to export data in its original form, that is, to request to not apply the redaction rules.




The following points describe the processes for exporting data:

- Some processes always export data as is, in case the target consumer needs the raw data. This is irrespective of the user requesting the export.
- Some processes perform data redaction by default and may offer an option for exporting data as is. The option to export data in its original form is secured by application security.
- In this example, process A supports data redaction by default with an option to export non-redacted data. User A has the right to export the data redacted (default) or explicitly request to export it in its original form. User B does not have the right to do so hence their export content is redacted by default.

The option to export data in its original form is associated with the **Do Not Apply Redaction Rules** access mode associated with the following relevant application services:

- For CMA, the additional access mode is associated with the migration data set export's application service. When granted, the user may select the special non-redacted export option when they create the export data set request.
- For GDE batch processes, the access mode is associated with their application service and is tied to the **Do Not Apply Redaction Rules** batch parameter they support. Only users with the proper application security can submit these batches using the special batch parameter option

# Defining Redaction Rules

This portal is used to view and maintain rules that define how specific fields that containing personal information should be redacted when exported out of the application or written to a debug log. For example, an e-mail address may be redacted by replacing some or all characters with '*'.

Refer to Data Redaction for Export and Logging for more information.

You can access the portal from the**Admin Menu**, and then **Security**, and then **Redaction Rule**. You are brought to a query portal with options for searching for a specific redaction rule. Once a redaction rule has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Redaction Rule.** This zone provides information about the selected rule.


Copyright © 2026, Oracle and/or its affiliates.

## Obfuscation Configuration

This portal is used to view and maintain all configuration options related to personal information obfuscation. It includes all masking, encryption and data redaction rules.

You can access the portal from the**Admin Menu**, and then **Security**, and then **Obfuscation Configuration**.

Refer to Encryption and Masking for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Redaction Rules.** This zone lists all redaction rules. You may add a new rule from this zone.
- **Masking.** This zone lists masking options. You may navigate to the masking related feature configuration to review and edit these options.
- **Encryption.** This zone lists encryption options. You may navigate to the encryption related feature configuration to review and edit these options.
- **Review Fields for Redaction Rules.** This zone lists fields that may contain personal identifiable information and are not associated with a redaction rule. Review these fields and add redaction rules when applicable.

# Cryptography Keys

The system provides support for defining various types of cryptography keys to handle different use cases in the system related to encrypting / decrypting data, providing credentials and providing and reading signatures when exchanging data between two parties. This may be used when communicating information with an external system or for internal processing.

The following sections include information about the functionality provided to support this type of security.

- Understanding Key Rings
- Defining Key Rings

## Understanding Key Rings

The Key Ring object is provided to define the specific use case and to reference the keys that are used over time for a given business use case. Each key ring references a specific business object, which defines all the rules related to managing the keys. For example:

- If the key ring is used to define a pair of private and public keys, then the business object includes actions for generating a key pair and for viewing the public key.
- If the key ring is used to capture the public key of a third party, then the business object includes actions for capturing the external public key.
- Each key ring defines the rules related to the possible lifecycle of the keys.

The following sections include information about the functionality provided to support different key ring classes for particular use cases.

- Field Encryption
- OAuth Keys
- RSA Signature Keys


Copyright © 2026, Oracle and/or its affiliates.

- File Encryption and Signing Keys

## Field Encryption

The system supports encryption of certain sensitive fields in the database. This support includes the ability to capture hash values of encrypted data for searching purposes.

Refer to Application Encryption to understand how to configure which fields you want to encrypt.

The sections in this topic provide more information about the defining the encryption and hash key values and the key rotation support.

### Symmetric Key

The product provides an AES Symmetric Key business object. It generates an encryption key using an AES 256 algorithm.

When defining a key ring for this business object, you provide a 3-character "wrap" prefix that is unique across key rings for the 'symmetric key' class. This aids in key rotation, described below.

The business object supports key rotation with respect to the keys. See the section below for more information about key rotation.

Once your symmetric key ring is defined and you have at least one active key, you can configure the fields that you want to encrypt and reference this key ring in the configuration. Refer to Application Encryption for more information.

### Hash Key

The product provides an HMAC Hash Key business object. It generates an HMAC key and stores it in PEM format.

When defining a key ring for this business object, you provide a 3-character "wrap" prefix that is unique across key rings for the 'hash key' class. This aids in key rotation, described below.

Once your hash key ring is defined and you have at least one active key, for encrypted fields that support a hash value, you can reference this key ring in the configuration. Refer to Application Encryption for more information.

### Key Rotation

Defining your encryption and hash keys using key rings allows for key rotation.

- When the system encrypts the data or creates the hash field, it uses a marker so that the system knows what key ring and key were used to encrypt / hash the data. Data is encrypted / hashed with the current Active key.
- If your organization wants to introduce a new key for either your symmetric key ring or your hash key ring, use the Generate Key button to create a new key. Click the Activate button adjacent to that Key to make it the key to use for all data that gets added or updated going forward. The previous key is marked Inactive.
- All records added or updated going forward use the new key.
- To rotate the keys across all the existing data encrypted / hashed with the previous key, you need to submit a batch. Refer to Mass Key Rotation for more information.


Copyright © 2026, Oracle and/or its affiliates.

### Key Ring Rotation

The respective business objects for the symmetric key ring and the hash key ring define the algorithms to use for generating the key. If there comes a point where you want to use a different algorithm for generating the symmetric encryption key or the hash key, a new business object is needed so that the new desired algorithm can be defined. The following points highlight the steps needed to change the key generation algorithm.

- Define the new business object for the use case (symmetric encryption key or hash key). Ensure that the algorithm related to the Generate Key state has the desired key generation logic. This business object should include a field for the 'wrap prefix', similar to the delivered product business objects for the symmetric key ring or hash key ring.
- Define a new key ring for the new business object and be sure to choose a different wrap prefix value. This is needed for key rotation.
- Generate a key for the key ring and activate it.
- Go to the Feature Option Configuration and update the keyRing or hashKeyRing references for all the entries to point to the new key ring.
- All records added or updated going forward use the new key ring and its active key.
- To rotate the keys across all the existing data encrypted / hashed with the previous key ring and its key, you need to submit a batch. Refer to Mass Key Rotation for more information.

## OAuth Keys

The system provides support for sending messages secured by OAuth authentication. To deliver the message, the application is required to provide a client ID and secret. The OAuth secret is sensitive data and needs to be stored securely in the application. The Message Sender configuration provides the ability to store the client ID and an encrypted secret directly in the sender context. Alternatively, the sender context can refer to a key ring that stores those details. Using a key ring allows users to apply new secrets over time without having to update the affected message sender configurations.

Refer to Message Sender - Context for more information about configuring a message sender for OAuth security.

The product provides an OAuth Keys business object that supports the following functions:

- Defining the external system related to the key ring
- Adding a client ID and secret key. The secret key is stored in an appropriate "secret store".
- Key rotation. For increased security, a new secret key may be entered periodically. The new key automatically becomes the active key.

## RSA Signature Keys

RSA signature key pairs are used in the following cases:

- Certain processes in the system may need to write files to or read files from Oracle Cloud Object Storage. Cloud object storage has its own method for encrypting files, and products using the framework application do not need to define additional configuration for that. However, when communicating with cloud object storage, the application is required to provide a signature key.

  Refer to External File Storage for more information about configuring the system to connect to cloud object storage.


Copyright © 2026, Oracle and/or its affiliates.

- Customers may need the ability to authorize a customer service representative (CSR) to view the customer's account information in a customer facing portal. A signature key is required to support secure communication with the endpoint.

  Refer to User Redirection for more information about configuring the system to enable this type of redirection.

- Signature keys are used for outbound messages when using the OAuth JWT assertion grant type. The key is used to sign the assertion.

The product provides an RSA Signature Key Pair business object that supports the following functions:

- Generating a key pair, public and private. The private key is stored in an appropriate "secret store" and the public key is available to copy and share with a third party.
- Viewing the public key, allowing a user to copy it to register it with cloud object storage.
- Activating the new key pair (to be done after registering the new public key).
- Inactivating an expired key.
- Key rotation. For increased security, a new key pair should be generated periodically.
- Deactivating all keys for the key ring.
- Requesting a Certificate Signing Request
- Certificate Signing Request

## Certificate Signing Request

If a certificate signing request is required for any use case, the following steps should be followed.

- Navigate to the signature key ring that requires a certificate.
- In the Keys zone, identify the pending or active key that requires a certificate. Click **Generate CSR**.
- You are prompted to provide appropriate distinguished name parameter values and set the duration.

> **Note**
>
> The distinguished name parameters are defined in the `F1-DistinguishedNameParameter` extendable lookup. The visible parameters are those that marked as 'default in grid'. Additional parameter names are available, if needed.

- Optionally check the Self Signed Certificate switch, for example in a development or test environment.
- Click Save

If you didn't check the self signed switch, copy the returned Certificate Signing Request text to a .pem file and use a certificate authority to provide the certificate. Otherwise, use the returned Self Signed Certificate.

Once you have the certificate, provide it to the relevant external party, according to the use case.


Copyright © 2026, Oracle and/or its affiliates.

# File Encryption and Signing Keys

For security, files exchanged with third parties may be encrypted, digitally signed or both. This requires the sender and recipient to maintain public/private key pairs and share public keys. The product supports the maintenance and generation of public and private keys in a format suitable for encryption using PGP standards.

The following sections provide additional information about encryption keys.

### Encryption Key Pairs

Internal encryption key pairs are used in the following cases:

- Creating a digital signature for an extract file. The signature is encrypted with the private key of the key pair and decrypted by the recipient using the shared public key .
- Decrypting an import file. The expectation is that the sender has encrypted the file data using a temporary key. The temporary key is encrypted using the shared public key and decrypted using the private key.

The product provides an Encryption Key Pair business object that supports the following functions:

- Generating a key pair, public and private. The private key is stored in an appropriate "secret store" and the public key is available to copy and share with a third party. Both keys are in formats suitable for PGP encryption.
- Viewing the public key, allowing a user to copy it to register it with third parties.
- Activating the new key pair (to be done after sharing the new public key).
- Inactivating an expired key.
- Key rotation. For increased security, a new key pair should be generated periodically.
- Deactivating all keys for the key ring.

### Public Encryption Keys

External public encryption keys are used in the following cases:

- Verifying the signature of an imported file. The signature is decrypted using the public key provided by the sender.
- Encrypting an extract file. The batch file adapters create a temporary key which is used to encrypt the file data. The temporary key is encrypted using the public key provided by the recipient

The product provides a Public Encryption Key business object that supports the following functions:

- Defining the external system and reference ID related to the key ring.
- Adding a public key to the key ring. The product expects the public key to be in a format suitable for PGP encryption.
- Viewing the public key.
- Inactivating an expired key.
- Key rotation. When new public keys are provided by the external system, the previously active key is set to expiring status. It may be manually inactivated after all messages using that key have been processed.


Copyright © 2026, Oracle and/or its affiliates.

- Deactivating all keys for the key ring.

## Defining Key Rings

Refer to Understanding Key Rings for an overview of key ring functionality.

To maintain the key rings applicable to your product or implementation, open **Admin**, and then **Security**, and then **Key Ring.**

This is an All-in-One portal and includes the standard List and display zones for a key ring.

The information captured on the key ring depends on the business objects supported by your product or implementation. Refer to the embedded help text for more information.

**Key Zones**

The key zones provide information about the keys that are associated with the key ring. The zones displayed and the available actions depend on the type of key. Refer to embedded help for more information.

**Key Rotation**

For internal keys, it is recommended to practice key rotation and generate new keys periodically. New key pairs are generated in the **Pending** state. The user should register the public key of the new key pair with the associated third party. Once that is done, the new key should be activated.

For external keys, new public keys are added in the **Active** state. The previously active key is set to **Expiring** and may be inactivated when no further processing needs to refer to it.

# User Interface Tools

This section describes tools that impact many aspects of the user interface.

- Understanding Portals and Zones
- Setting Up Portals and Zones
- Defining Menu Options
- Defining Navigation Keys
- Defining Navigation Options
- Navigation to External Pages
- Understanding System Messages
- Understanding Unified Search
- Trees
- Contextual Insights
- Knowledge Articles
- Color Configuration
- Defining Display Icons
- Custom Look and Feel Options


Copyright © 2026, Oracle and/or its affiliates.

# Understanding Portals and Zones

A portal is a page that is comprised of one or more information zones. The base product pages are built using either a fixed page metaphor or using portals and zones.

There are three broad classes of portals:

- **Standalone.** Standalone portals are separate pages where the main tab of the page is built using a portal. These pages are opened using any of the standard methods (e.g., by selecting a menu item, by selecting a favorite link, etc.). Additional tabs for a stand-alone portal may be included using tab page portals.
- **Tab Page.** These types of portals cannot be attached to a menu. They simply define the zones for a tab on a standalone portal.
- **Sidebar.** A sidebar portal is a special portal that dictates what may appear in the Sidebar Area. Its zones contain tools and information that exist on the user's desktop regardless of the transaction. Several sidebar portals and zones are delivered as part of the base-package. Your implementation can add additional zones to these portal.
- **Assistant.** This is a special portal that defines what may appear in the Assistant Area if configured by your specific product.

The contents of this section describe general information about portals and zones.

### Portal Page Structure

A portal page has a main tab and may have additional tabs. The page is constructed by first creating a **Standalone** portal, which is also the main portal, followed by creating a **Tab Page** portal for each additional tab and linking it to the main portal. Content is organized on each tab using independent zones.

The entire page is secured by the single application service associated with the main portal.

### Portal Roles

An optional attribute called a portal role may be assigned to a portal. The role is typically for information purposes, but in some cases, a role may support additional configuration.

Certain roles may support predefined **UI Template** layouts to choose from.

### Portals Are Made Up of Zones

A portal is a page that contains one or more zones, and each zone contains data of some sort. All zones reference a **Zone Type**. The zone type controls the behavior of the zone and the parameters available to configure the zone.

A portal may also use zones to present additional information in a drawer. These zones are not part of the page layout and are only presented upon a user request.

### Not All Zones Make Up Portals

Zones may or may not be associated with a portal. The following describes additional uses of zones:

- A data explorer type of zone may implement a search for an entity on an input user interface map. These zones may be referenced on a FK reference record or directly referenced on a UI map.


Copyright © 2026, Oracle and/or its affiliates.

- A data explorer type of zone may implement a Business Service. The latter may be used for an internal process or used to build a user interface dropdown.
- A zone may be defined as a **Context Sensitive Zone** that appears on the Sidebar when a specific page or portal is in context. This type of zone is not linked to the Sidebar portal but associated with the context page or portal's navigation key.
- A 'view detail' zone may be launched into a drawer without being configured as part of the portal it is launched from.

### Configuring Zones for a Portal

The portal includes configuration of how the zones should appear on the portal by default. This includes the following options, all of which may be overridden by an implementation.

- The order in which the zone should appear. An implementation may configure an override sequence to change the order zones on a base delivered portal.
- Whether the zone is visible on the portal. Zones delivered in the base product should be configured to be visible. But an implementation may override this if desired.
- Whether the zone should display initially collapsed or not. A zone's data is only retrieved when it is expanded. As such, a zone may be configured to be initially collapsed when the data is not needed very often. A user can expand the zone when the information is required. Implementations may change the collapsed setting of a base product portal / zone. Refer to Zones May Appear Initially Collapsed When a Page Opens for more information.

### Flexible Layout Control

By default a zone spans the full width of the portal's display area but can be configured in a number of ways using the layout control features on the portal.

These features allow portal designers to have finer-grained control over how zones are laid out, and enables them to achieve a better user experience and a more optimal layout that reduces unused space.

The following highlight key layout control options:

- In addition to the standard half and full width options, 1/4, 3/4, 1/3 and 2/3 width options are also available.
- Better handling of zones that are dynamically hidden by a visibility script. When a zone becomes invisible its space may be reclaimed by a next zone as long as it is of same width or smaller.
- A zone may define the beginning of a new row. This indication allows for a more fixed layout that is not impacted by hidden zones. The new row indication may establish a layout of one, two or three zones displayed vertically.
- Support the following zone height options:
  - **Content (Unlimited Height)** lets the height be determined by content.
  - **Content (Limited Height)** lets the height be determined by content up to a specified maximum limit. This option may be used to achieve a more consistent portal layout, especially when a zone is rendered on long or small devices.
  - **Fixed Height** sets the zone at a specific height regardless of content.

    This option too may be used to achieve a more consistent portal layout.
  - When a height option is not explicitly selected, zone height is unlimited and determined by content. However, for a data explorer zone, the results grid area itself (not the entire


Copyright © 2026, Oracle and/or its affiliates.

zone) is limited by the zone parameter "height of report" (which has a default value of 50%).

You may also control the visibility of the following page components:

- **Page Title.** Use the **Page Title Visibility** portal option on a standalone portal to indicate that the page title should not be visible for this portal and its related tab portals if any.

  > **Note**
  >
  > The page actions displayed in the same row as the page title are also suppressed. As such, you should only use this option if the actions in that area are not needed, including Bookmark, Clear and Refresh.

- **Tab Menu.** Use the **Tab Menu/Tab Visibility** portal option to suppress one or more tabs on the page based on conditions. It also may be configured to suppress the tab area altogether.
- **Zone Header.** You may use the **Zone Header Visibility** general zone parameter to suppress an explicit map zone's header area to allow for more vertical space when the page only has one zone and the standard actions on the zone header are not relevant.

### Portal Specific Page Actions

The system supports defining actions that appear in the Page Actions Toolbar that are specific to a portal. Portal specific actions are first defined using a menu with the menu type of **Page Action Menu**. Each menu line in a page action menu should define one item. The menu line's item may reference a navigation option or a sub-menu (which in turn defines one or more lines with menu items that define navigation options). Use the **Page Action Menu** portal option to associate the menu with the portal. More than one page action menu option may be defined. If the lines on the referenced menus define navigation options, they are shown as distinct buttons. If any line references a sub-menu, then an action menu is shown, using the description on the menu line.

> **Note**
>
> Standard menu security is checked. For any menu item in any of the page action menus defines an application service / access mode, you only see the action if you have security.

### Advanced Configuration for Portal Title / Tabs

You may configure your portal to provide additional information to be displayed either in the portal's Page Title Area or in the tab area.

**Information String**. If the portal is configured to display a single record, you can configure the **Page Info FK Reference** portal option to include the FK reference for the related maintenance object. The system will automatically display the information string for the FK reference when viewing a record on this type of portal.

**Advanced Title Configuration**. You may provide a service script in the **Enhanced Page Title** portal option to override the page title or to add additional information to be displayed. The service script should include the data area F1PageTitleDetails. Refer to the detailed description of the data area for more information on what the service script can return. This option may only be configured on the standalone ("main") portal, but the information is visible on any of the portal tabs.


Copyright © 2026, Oracle and/or its affiliates.

**Scorecard Configuration**. For portals that are configured with the role of Dashboard, you may configure a scorecard for each tab in the portal. To do this, you provide a service script in the **Scorecard** portal option. The service script should include the data area F1ScorecardDetails. Refer to the detailed description of the data area for more information on what the service script can return. If the dashboard has multiple tabs, if any of the tabs configure a scorecard service script, all tabs must configure one.

### Portal Banners

Important messages related to the context of a page may be displayed in the form of banners presented in the title area. For example, a customer dashboard may show a banner message alerting the agent of an existing outage along with the ability to view outage details in a drawer.

These messages are composed by a script referenced on a **Portal Banners** portal option. Banner messages may be associated with various severity levels and optionally support related actions. The option is defined for the main tab of the portal and applies to the entire page.

The product provides a generic 'Portal Banners by Insight Class' (F1BannesInst) script that composes messages based on insights for an insight class and a given context. Using this script allows banners for a portal to be configurable by the customer.

### Portal Templates

When defining a portal with the portal role of **Dashboard**, you may optionally choose a **UI Template**. The UI templates define a predefined layout. You simply need to choose the zones that you wish to include and the template will render the zones based on the template's configuration.

### User Preferences

A **Standalone** portal includes configuration to indicate whether or not the portal should appear on a user’s portal preferences. The portal will not appear on the user's portal preferences if the **Show on Portal Preferences** setting is set to No. Note that an implementation may change this value for a product delivered portal. This setting is typically enabled for a portal that provides disparate information where not all zones are applicable to all users or where users may wish to adjust the order of the zones. An example of a portal enabled for portal preferences is the Sidebar portal.

The user can override zone oriented configuration for the portal:

- Which zones appear on that portal
- The order in which the zones appear
- Whether the zones should be initially collapsed when the portal opens.
- The refresh seconds. This is applicable to zones displaying data that changes often.

An implementation can optionally configure the system to define portal preferences on one or more "template" users. When a template user is linked to a "real" user, the real user's preferences are inherited from the "template" user and the "real" user cannot change their preferences. Some implementations opt to work this way to enforce a standard look and feel for users in the same business area.

Refer to User — Portal Preferences for more information about how users configure their zones.


Copyright © 2026, Oracle and/or its affiliates.

### View Detail Drawers

A zone may be designed to display related extended information in a drawer. This type of zone is designed and implemented like any other zone. It is recommended to associate it with the 'View Details Drawer' role for reporting purposes.

View detail zones can be configured as follows:

- **Portal panels.** An optional list of 'view detail' zones can be defined as part of a portal's zone configuration. These zones are not rendered as part of the portal's layout. Instead, they represent groups of zones to be presented in a drawer upon user request. A 'view detail' zone is associated with a **Panel Name** which represents a uniquely named drawer content. When the panel name is launched all zones that belong to that panel are presented in the drawer area in sequential order.
- **Not related to a portal.** A zone can be launched directly into a 'view detail' drawer, without being listed on any portal. This technique is used when the zone displays information that is not specific to the current portal. For example, the Knowledge Article drawer may be launched from any portal that supports a default knowledge article.

The product supports small, medium and large drawer sizes. When launched as a panel, the largest drawer size setting of all panel zones is used. When a zone is directly launched, not via an association to a panel, the width of the zone sets the drawer width.

A drawer may be launched from another drawer, forming a nested display of panels.

Support for launching a drawer is available from a display UI map, a data explorer column, and a chart. Refer to Launch a View Details Drawer for more information.

> **Note**
>
> Drawers may not be launched from the Sidebar and Assistant areas.

### View Entity Drawer

In some pages, the product supports the ability to review an entity's information in a drawer without navigating to the entity's page.

The default drawer for viewing an entity's information may be dynamically determined using the **View Entity Drawer** maintenance object or business object option. If a business object option exists it overrides the maintenance object option if any. When applicable, you may use these options to override the 'view details' zone used by the product an entity. To do so, implement your custom 'view detail' zone and reference it as an option with a higher sequence number.

This feature is only available in select pages. By default, a link to an entity navigates to its portal, and does not launch the entity's view details drawer. To implement such functionality in a custom user interface, refer to product examples that use the 'Get Drawer Zone for an Entity' (F1GetDrwEnt) script.

### Removing Zones From Portal

You cannot remove a base product zone from a base product portal. An implementation may override the **Display** setting to prevent a zone from displaying on the portal. In addition, you cannot remove a zone if a user has enabled it on their portal preferences. To remove a zone from the portal list, first make sure that no user has it enabled in their portal preferences.


Copyright © 2026, Oracle and/or its affiliates.

### Granting Access to A Portal

An application service is associated with each portal. A user must be granted access rights to the respective application service in order to see a portal.

When you add a new **Standalone** portal, the system automatically creates its application service behind the scenes and displays it as part of the portal definition. You'll need to use this application service to grant access to the portal. **Tab Page** portals do not have separate security access. If a user has access to the main stand-alone portal, then the user will have security access to all its tabs.

Please note the following in respect of how application security impacts a user's portals:

- A user's Portal Preferences page only shows the portals configured to show on user preferences and where they have security access.
- The system's menus only show portals to which a user has security access.
- Users can set up favorite links to all portals, but they must have security rights to the portal's application service in order to invoke the favorite link.

### Granting Access to Zones

An application service is associated with each zone. A user must be granted access rights to the respective application service in order to see a zone on a portal. Refer to The Big Picture Of Application Security for information about granting users access rights to an application service.

Please note the following with respect to zone application security:

- For most base product portals, all the zones for all the portals reference the same application service that is used to grant access to the portal. In other words, if the user has access to the page, then the user has access to all the zones on all portals for the page. There may be exceptions to this rule for certain portals.
- For a base product multi-query zones, typically the individual query zones and the multi-query zone reference the same application service that is used to grant access to the main (standalone) portal for the page. However, there may be individual query zones provided with a unique application service. This may occur when the query option is unusual and not applicable to all users or even to all implementations. If a user does not have security access to an individual query zone, that option will not be available in the dropdown.
- For base product portals that are configured to show on portal preferences, it is common that the portal contains different types of zones that may be applicable to different types of users. Typically these types of portals will deliver a unique application service for each zone so that an implementation may configure which user groups are allowed to view each zone. For these types of portals, please note the following:
  - A user's Portal Preferences page contains a row for a zone regardless of whether the user has access rights to the zone. Because of this, the system displays an indication of the user's access rights to each zone.
  - If a user's access rights to a zone are revoked, the zone will be suppressed when the user navigates to the respective portal.
  - Revoking a user's access rights does not change the user's portal preferences (i.e., a user can indicate they want to see a zone even if they don't have access to the zone - such a zone just won't appear when the respective portal appears).


Copyright © 2026, Oracle and/or its affiliates.

#### Putting Portals on Menus

When you add a new **Standalone** portal, the system automatically creates its navigation option behind the scenes and displays it as part of the portal definition. You'll need to use this navigation option to add the portal to a menu or to allow any other type of navigation method to the portal.

To add a portal to a menu, you must add a menu item to the desired menu and reference the portal's navigation option on the new menu item. A portal's navigation option can appear on any number of menu items (i.e., you can create several menu items that reference the same portal).

Your users can set up their preferences to include the portal's navigation option on their Favorite Links. This way, they can easily navigate to the portal without going through menus.

#### Portal Options

Portal options are used to enable various features available for each portal. Several are mentioned in previous sections in this topic. Review the detailed descriptions of each portal option to understand what other features are available.

> **Note**
>
> **You can add new option types.** Your implementation may want to add additional portal option types. To do that, add your new values to the customizable lookup field `PORTAL_OPT_FLG`.

# Setting Up Portals and Zones

The topics in this section describe how to set up portals and zones. Please refer to The Big Picture of Portals and Zones for background information.

- Defining Zone Types
- Defining Zones
- Defining Context-Sensitive Zones
- Defining Portals

## Defining Zone Types

A Zone Types represents a particular type of zone with a specific behavior. For example, a data explorer zone type is used to select data using a specific SQL statement and display the data based on parameter configuration. The zone type defines the Java Class that controls the behavior of the zone and defines the parameters that the Java Class supports. The base product supports many zone types used to build the portal / zone user interface. Implementations may introduce their own zone types.

> **Note**
>
> It is not very common for an implementation to introduce their own zone types.

Select **Admin**, and then **System**, and then **Zone Type** to maintain zone types.


Copyright © 2026, Oracle and/or its affiliates.

### Description of Page

Specify an easily recognizable **Zone Type** code and **Description**. Use the **Detailed Description** to describe in detail what the zone type does.

> ⚠ **Caution**
>
> When adding new zone types, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

**Owner** indicates if this zone type is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a zone type. This information is display-only.

**Java Class Name** is the Java class responsible for building the zone using the parameters defined below.

Two types of parameters are specified when defining a zone type:

- Parameter values that have a **Usage** of **Zone** are defined on the zones and control the functionality of each zone governed by the zone type. A **Usage** value of **Zone - Override Allowed** indicates that an implementation may override the parameter value for a base zone.
- Parameter values that have a **Usage** of **Zone Type** are defined directly on the zone type and control how the zone type operates (e.g., the name of the XSL template, the name of the application service). A **Usage** value of **Zone Type - Override Allowed** indicates that an implementation may override the parameter value for a base zone type.

The following points describe the fields that are defined for each parameter:

- **Sequence** defines the relative position of the parameter.
- **Parameter Name** is the name of the parameter.
- **Description** is a short description that allows you to easily identify the purpose of the parameter.
- **Comments** contain information that you must know about the parameter or its implementation. For parameters with a usage of **Zone** or **Zone - Override Allowed**, this information is visible to the user when viewing or defining this parameter for a zone of this type.
- **Usage** indicates whether the parameter value is defined in a **Zone** of this type or in the **Zone Type**. **Zone - Override Allowed** and **Zone Type - Override Allowed** indicate that override values for the parameters defined in a base zone or base zone type can be entered.
- **Required** is checked to indicate that a zone must define a value for the parameter. It is not checked if a value for the parameter is optional. This field is protected if the **Usage** is **Zone Type** or **Zone Type - Override Allowed**.
- **Parameter Value** is used to define the value of zone type parameters. This field is protected if the **Usage** is **Zone** or **Zone - Override Allowed**.
- **Owner** indicates if this parameter is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a parameter. This information is display-only.


Copyright © 2026, Oracle and/or its affiliates.

### Zone Type Parameter Comments

For the product owned zone type parameters, the parameter's detailed description provides the detail needed for properly configuring the parameter. For the Action parameters (IMPLEMENTOR_ACTION_n), the parameter description is abbreviated. Additional detail about configuring this parameter may be found in the Zone Action Parameter detailed information. The same details apply.

## Defining Zones

This portal is used to view and maintain the configuration of a zone. Implementations may define custom zones or override the descriptions and some parameter values for base product zones.

You can access the portal from the**Admin Menu**, and then **System**, and then **Zone**. You are brought to a query portal with options for searching for a specific zone. Once a zone record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Understanding Portals and Zones for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Zone.** This zone provides general information about the selected zone.
- **Query Options.** This zone maintains the list of query options for a Multi-Query zone. The zone is only visible for Multi-Query zones.
- **Filters.** This zone maintains various zone parameters that handle filtering options for a data explorer zone. The zone is only visible for data explorer zones.
- **General Parameters.** This zone maintains various general purpose zone parameters for a data explorer zone.
- **SQLs.** This zone maintains individual SQLs and their related parameters. The zone is only visible for data explorer zones.
- **SQL Configuration.** This zone maintains all the zone parameters related to a specific data explorer SQL. The zone is only visible when a specific SQL is broadcast from the **SQLs** zone.

The following zones may appear as part of the portal's **References** tab page

- **Zone References.** This zone list various configuration entities referencing the current zone.
- Overriding Parameter Values
- Overriding Zone Parameters
- Zone Parameter Details
- Data Explorer Graph Configuration
- Pagination Configuration


Copyright © 2026, Oracle and/or its affiliates.

- Auto-filtering Configuration
- Use Data Explorer for Derived Data
- Configuring Timeline Zones

## Overriding Parameter Values

Implementation teams may override the value of selected types of base provided zone parameters. The override parameter value field for base owned zone parameters is only enabled when the related zone type parameter usage is **Zone - Override Allowed**.

For multi-query zones, you may use the override parameter value to add custom query options to a base product owned zone.

## Overriding Zone Parameters

Implementations may override the value of a base owned zone parameter only if the zone type definition allows such override. The override parameter value is enabled for a specific base owned parameter when the related zone type parameter usage is set to **Zone - Override Allowed**.

You may use the override parameter capability to provide alternative or additional custom query options for a base owned multi-query zone.

## Zone Parameter Details

For most zone parameters, the embedded help for the parameter provides the detailed information needed for configuring the parameter values. For some parameters with very detailed descriptions, the embedded help is abbreviated and more detail is provided here.

- Zone Visibility Service Script
- SQL Statement
- Column Parameters
- Zone Action
- User Filters
- Hidden Filters
- Multi-Select Action

### Zone Visibility Service Script

All zones support a visibility script that is used to determine if the zone should be displayed to the user or not based on conditions. The script may receive input parameters and is expected to return a Boolean value indicating if the zone should be displayed or not. The embedded help for the **Zone Visibility Service Script** parameter provides details related to the syntax.

The following table highlights some service scripts provided by the product that may be used if applicable to your zone's requirements. This is not an exhaustive list of visibility scripts. There may be others that are specific to a given zone.

| Script Code | Description | Comments |
|---|---|---|
| `F1-ShldShwZn` | Zone Visibility - Display Zone in Portal | This script simply returns a value of 'true' and is used when the zone should always appear. |


Copyright © 2026, Oracle and/or its affiliates.

| Script Code | Description | Comments |
|---|---|---|
| F1-CondShwZn | Zone Visibility - Display Zone in Portal Conditionally | This is used when the condition for showing the zone is based on the population of a context value. This is commonly used when one zone in the portal should only appear after a broadcast of a record from another zone in the portal.<br>For an example of a zone the uses this visibility script, refer to F1-BSFTYPE . |
| F1-RwCtShwZn | Zone Visibility - Based on Row Count | This is used when the condition for showing the zone is based on the existence of one or more rows that can be determined using SQL. This script accepts a zone code, user filters 1 through 25 and hidden filters 1 through 10. The script returns an indication of 'true' if at least one row count is returned by the zone.<br>To use this visibility script, a specific data explorer zone must be developed for the specific use case.<br>For an example of a zone the uses this visibility script, refer to F1-MIGRREQEL . |

## SQL Statement

Data explorer zones are used to select data to display using one or more SQL statements. The SQL parameters are applicable to the following zone types

- Info Data Explorer - Single SQL (F1–DE-SINGLE). The parameter has the description **SQL Statement**.
- Info Data Explorer - Multiple SQLs (F1–DE). The parameters follow the description pattern of **SQL Statement x**.
- Query Data Explorer - Multiple SQLs (F1–DE-QUERY). The parameters follow the description pattern of **SQL Statement x**.

> **Note**
>
> If your implementation has been configured to restrict the functions that may be used when defining an SQL then an error is issued at runtime if there are functions found that are not in the allowlist. The allowlist may be viewed using the **SQL function allowlist** link in the SQLs zone on the zone maintenance page.

> **Note**
>
> Zone SQL should take care not to use the following reserved keywords for the table alias. (For example, the table alias in this SQL is 'XX': SELECT 'X' FROM W1_ACTIVITY_TYPE **XX**.)

| Keyword |
|---|
| **AT** |


Copyright © 2026, Oracle and/or its affiliates.

The following table provides a list of SQL substituted keywords that may be used in the SQL Statement parameters in explorer zones. At execution time, the system determines the database and substitutes the keyword with the database specific syntax:

| Keyword | Description | Examples |
|---|---|---|
| `@toCharacter()` | Converts the input to Character data type. | select **@toCharacter**(batch_cd) as batchCode from ci_batch_ctrl |
| `@toDate()` | Converts the input to Date data type. | select **@toDate**(last_update_dttm) as lastUpdateDate from ci_batch_ctrl |
| `@toNumber()` | Converts the input to Number data type. | select **@toNumber**(next_batch_nbr) from ci_batch_ctrl |


Copyright © 2026, Oracle and/or its affiliates.

| Keyword | Description | Examples |
| --- | --- | --- |
| `@currentDate` | Fetches the current date. | select batch_cd, **@currentDate** as today from ci_batch_ctrl |

> **⚠ Caution**
>
> The Oracle functions **SYSDATE** and `CURRENT_DATE` should not be used because they do not properly cater for adjusting dates from the database time zone to the installation time zone, if needed.


Copyright © 2026, Oracle and/or its affiliates.

| Keyword | Description | Examples |
| --- | --- | --- |
| `@currentTimestamp` | Fetches the current date / time.<br><br>**⚠ Caution**<br>The Oracle functions **SYSTIMESTAMP** and `CURRENT_TIMESTAMP` should not be used because they do not properly cater for adjusting the date / time from the database time zone to the installation time zone, if | select batch_cd from ci_batch_ctrl where last_update_dttm > **@currentTimestamp** |


Copyright © 2026, Oracle and/or its affiliates.

| Keyword | Description | Examples |
|---|---|---|
| | need ed. | |
| `@concat` | Combines the result list of two or more columns. | select batch_cd **@concat** next_batch_nbr concatNbr from ci_batch_ctrl |
| **@substr(string, start)** | **String** is the input String that you are trying to get a substring of.<br>**Start** is the position of the character for the output results. | select batch_cd batchCode from ci_batch_ctrl<br>Result: TESTCD<br>select **@substr**(batch_cd,3) batchCode from ci_batch_ctrl<br>Result: STCD |
| **@substr(string, start, end)** | **String** is the input String that you are trying to get a substring of.<br>**Start** is the position of the character for the output results.<br>**End** is the number of characters required in the output from starting position. | Select batch_cd batchCode from ci_batch_ctrl<br>Result: TESTCD<br>select **@substr**(batch_cd,3,2) batchCode from ci_batch_ctrl<br>Result: ST |
| `@trim` | Trims the white spaces of the output on both sides. | select **@trim**(batch_cd) as batchCode from ci_batch_ctrl |
| **The following syntax is related to 'fuzzy' searching. It is only applicable if Oracle DB Text is enabled and a context text index has been created. Refer to Advanced Search Options for more information.** | | |
| **@fuzzy(string, score, numresult, 'weight')** | **String** is the input value for the search.<br>**Score** is the degree of 'fuzziness'. Valid values are between 1 - 80. The higher the number the more precise the search. Default is 60.<br>**Numresults** is the number of variations to consider for the string. Valid values are between 1 and 5000. Default is 100.<br>Indicate `'weight'` to signal that the results are returned in order of weight. Leave this setting off to indicate that the results are returned in order of score. | Set score to 70, number results to 6, and specify weight.<br>select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1,70, 6, 'weight')) > 0 |
| `@fuzzy(string)` | This returns a string result from the fuzzy expansion operation where the default value of 60 is assumed for the **score** and the default value of 100 is assumed for the **numresult**. | To use default values:<br>select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1))> 0 |


Copyright © 2026, Oracle and/or its affiliates.

| Keyword | Description | Examples |
|---|---|---|
| **@fuzzy(string, score)** | This returns a string result from the fuzzy expansion operation with the **score** specified and the default value of 100 for the **numresult**. | Set score to 70.<br>select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1,70)) > 0 |
| **@fuzzy(string, score, numresult)** | This returns a string resulted from the fuzzy expansion operation with the similarity score and the **numresults** specified. | Set score to 70, number results to 6.<br>select user_id, last_name from sc_user where contains(last_name, @fuzzy(:F1,70, 6)) > 0 |

## Column Parameters

Data explorer zones are used to select data to display using one or more SQL statements. For each SQL statement, the zone may configure up to 20 Columns that contain the formatting definition for displaying the output data.

These parameters are applicable to the zone types

- Info Data Explorer - Single SQL (`F1-DE-SINGLE`). The parameters follow the description pattern of **Column x**.
- Info Data Explorer - Multiple SQLs (`F1-DE`). For this zone type, all SQLs are executed and the zone displays a union of all the results. The parameters follow the description pattern of **Column x for SQL y**. There are some mnemonics that do not make sense to differ within rows of the same column. For example, the column label. For these mnemonics, only the value in SQL 1 is considered for that column. The table below indicates which mnemonics follow this rule.
- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`). For this zone type, only one SQL is executed. The **SQL y Condition** parameter may be used to control this. The system will execute the first SQL whose condition is satisfied (or with no condition populated). The parameters follow the description pattern of **Column x for SQL y**.

The following sections describe the various types of mnemonics.

### Source Mnemonics

This table describe the mnemonics that control how the data in a column is derived.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `source=` | Defines how the column's value is derived. | `SQLCOL` | Indicates that the source of the column's value comes from a column in the SQL statement. This type of column must also reference the `sqlcol=` mnemonic. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | BO | Indicates that the source of the column's value comes from a business object. This type of column must also reference the `bo=`, `input=` and `output=` mnemonics to define how to interact with the business object. |
| | | BS | Indicates that the source of the column's value comes from a business service. This type of column must also reference the `bs=`, `input=` and `output=` mnemonics to define how to interact with the business service. |
| | | SS | Indicates that the source of the column's value comes from a service script. This type of column must also reference the `ss=`, `input=` and `output=` mnemonics to define how to interact with the service script. |
| | | FORMULA | Indicates that the source of this column's value is calculated using a formula. This type of column must also reference the `formula=` mnemonic. |
| | | SETFUNC | Indicates that the source of this column's value is calculated using a superset of values from the rows in the SQL statement. This type of column must also reference the `setfunc=` mnemonic. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | ICON | Indicates that the source of this column's value is a display icon reference (meaning that an icon will be displayed in the column). This type of column must also reference the icon= mnemonic to define the icon reference. |

**Note**

When using this source mnemonic, the format


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | ngmnemonic type = isnotapplicable. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | FKREF | Indicates that the source of this column's value is an FK reference (meaning that the FK reference's context menu and information string will be displayed in the column and will be enabled for navigation). This type of column must also reference the `fkref=` and `input=` mnemonics to define how the FK reference is called. |

**Note**

When using this source mnemonic, the


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|

formattingmnemonic type=isnotapplicable.

**Note**

To show af


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | oreignkey'sinformationbutnotenablenavigation,referto Comm |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | on Use Cases for the recommended technique. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | SPECIFIED | Indicates that the source of this column's value is specified by concatenating literals and other column values. This type of column must also reference the spec= mnemonic.<br>When constructing an HTML element for a web component or a function, the syntax may be slightly different for a data explorer column. Refer to UI Map Attributes and Functions for the appropriate syntax. |
| | | MSG | Indicates that the source of this column is a message from the message table (along with any substitution variables). This type of column must also reference the msg= mnemonic. |
| sqlcol= | Defines the column in the SQL statement when source=SQLCOL. | COLUMN_NAME | Enter the name of a column that is retrieved in the SELECT statement. Note that if the select statement uses an alias for a column, then the alias should be referenced here. |
| | | x | Where x is an integer value that references a column by its relative position in the SELECT statement. For example, sqlcol=3 would display the 3rd column in the SELECT statement). |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| bo= | Defines the business object to invoke when source=BO.<br>This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to / received from the business object. | 'Business Object Code' | |
| bs= | Defines the business service to invoke when source=BS.<br>This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to/received from the business service. | 'Business Service Code' | |
| ss= | Defines the service script to invoke when source=SS.<br>This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to / received from the service script. | 'Service Script Code' | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `icon=` | Defines the icon to use when `source=ICON`. | Cx | This means the Display Icon code is defined in an earlier column. For example, define `C1` if column 1 defines the icon code value. |
| | | COLUMN_NAME | This means the Display Icon code was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. |
| | | 'image' | This means the image file is defined directly. For example '/images/treeRootObject.gif'. |
| | | 'Display Icon Code' | This means the Display Icon code is defined directly. For example 'F1EDIT. |
| `fkref=` | Defines the FK reference used to retrieve the column's information when `source=FKREF`.<br>This mnemonic must be used in conjunction with the `input=` mnemonic to define how information is sent to the FK reference to build the information. | Cx | This means FK reference code is defined in an earlier column. For example, define `C1` if column 1 defines the FK reference value. |
| | | COLUMN_NAME | This means the FK reference was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. |
| | | 'FK Reference Code' | This means the FK Reference is defined directly. For example 'F1-ROLE'. |
| `formula=` | Defines the formula to use when `source=FORMULA`.<br>Examples:<br>• `formula=C1*.90/C2` | The formula can contain numeric constants, operators and column references. | For column references, use the format Cx where x represents the column number. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | • `formula =(C1/ C2)*100` | | Refer to Expression Parser for information about the functions supported. |
| `setfunc=` | Defines the function to apply to the rows of a given column when `source=SETFUNC`. | function(Cx) | Where Cx represents a column whose rows should have the function applied and the function is one of the following:<br>• **MAX**. This derives the maximum value of all rows in the column.<br>• **MIN**. This derives the minimum value of all rows in the column.<br>• **TOT**. This derives the sum (total value) of all rows in the column.<br>• **ACC**. This derives the cumulative total of all rows up to an including the current row. |
| `input=` | This is used to define one or more input fields and values passed to business objects, business services, service scripts, and FK references. The syntax is as follows: [ELEMENT_NAME=ELEMENT_REF ELEMENT_NAME=ELEMENT_REF ...] In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the | Cx | Where Cx represents the value of a previous column. If the value to pass is in the first column, reference C1. |
| | | COLUMN_NAME | This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. |
| | | 'literal value' | This means a literal value within the single quotes should be passed in. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | ELEMENT_NAME, which is the name of the element / field in the target. ELEMENT_REF is the value passed in. The next column indicates the possible values for ELEMENT_REF. | `userTimeZone` | This means the current user's time zone should be passed in. This is typically used with the business service `F1-ShiftDateTime` to convert data in the storage time zone to the user's time zone for display. |
| | | `installationTimeZone` | This means the installation time zone should be passed in. This is typically used with the business service `F1-ShiftDateTime` to convert data in the storage time zone to the installation time zone for display. |
| | | Examples:<br>• `input=[USER_ID=C1]`<br>• `input=[USER_ID=USER_ID]`<br>• `input=[input/targetTimeZone=userTimeZone]` | |
| `output=` | This is used to define the name of the element retrieved from the business object, business service or service script used to populate this column. | elementName | Example: `output=personInfo` |
| `pagingkey=` | This mnemonic is only applicable when the **Enable Paging** parameter has been configured. It indicates that this column is one of the keys used by the SQL statement to orchestrate paging through | `Y` | |

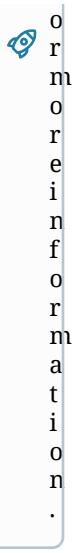

Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | results. This mnemonic can only be specified when the `source=SQLCOL`. | `N` | This is the default, meaning that you don't need to indicate `pagingkey=N` at all to indicate that the column is not one of the paging keys. |

**NOT_SUPPORTED**

Refer to PaginationConfiguration f


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | |  | |

> **Note**
>
> If multiple columns are configured with the same source BO, BS or SS and the same input data, the system caches the output from the first call and reuses the results for subsequent columns.

### Formatting Mnemonics

This table describe the mnemonics that control how a column is formatted.

> **Note**
>
> For the `F1-DE` zone type, zone displays the union of all the different SQLs.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `type=` | Defines how the column's value is formatted. | `STRING` | Columns of this type capture a string. This is the default value. |
| | | `DATE` | Columns of this type capture a date and will be displayed using the user's display profile. |
| | | `TIME` | Columns of this type capture a time (in database format) and will be displayed using the user's display profile. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | `DATE/TIME` | Columns of this type capture a date and time (in database format) and will be displayed using the user's display profile. |
| | | `MONEY` | Columns of this type capture a monetary field. This type of column may also reference the `cur=` mnemonic. If the cur mnemonic is not specified, the currency code on the installation record is used. |

**on and Foreign Key columns.** The source=sourcemnemonic may be use


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | NUMBER<br>dtoindicateacolumnshouldbederivedfromanicon referenceoraf | Columns of this type capture a numeric field. This type of column may also reference the dec= mnemonic. |

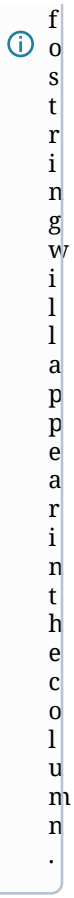

Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | |  oreignkey(FK)referenceIfyouuseeitheroftheseresources,thet | |


Copyright © 2026, Oracle and/or its affiliates.

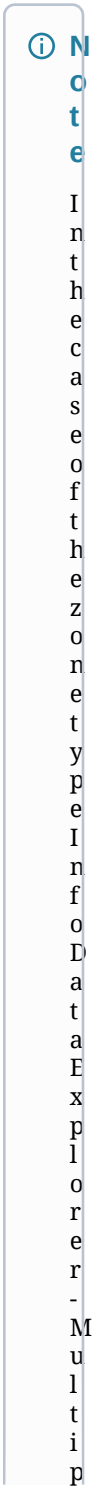
| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | ype=mnemonicisnotrelevantaseitheranicononracontextmenu/in | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | |  | |
| label= | Defines the column's override label. The label appears in the column's heading and in the zone's drag and drop area. If this mnemonic is not defined, the system uses the column's default label. The source of a column's default label differs depending on the column's source. Note that some sources don't have a default | FIELD_NAME | Enter a valid field name whose label should be used for the column label. This should always be the option used if multiple languages are needed. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | value and omitting this mnemonic will result in a blank label. | 'text' | Defines the text directly. |

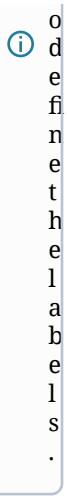




Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | leSQLs(F1_DE), thesystemwillusethecolumndefinitionsinSQL1t | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | |  odefinethelabels. | |
| `cur=` | Defines the currency code applied when `type=MONEY` if the installation record's currency should not be used. | Cx | This means currency code value is defined in an earlier column. For example, define `C1` if column 1 defines the currency code. |
| | | COLUMN_NAME | This means the currency code was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. |
| | | 'Currency Code' | This means the currency code is defined directly. For example 'USD'. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| dec= | Defines the number of decimal places when type=NUMBER.<br>It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record. | nR | Where n is the number of decimal places to show. Suffixing the number of decimal places with R means that the system should round up / down. Simply specifying n (without an R) means that decimal places should be truncated. For example, entering dec=4 will display 4 decimal places and truncate the remainder. |

**Note**

**Formatting only.** This mnemonic is only


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | used for formatting, it does not impact the precision used for subs |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | equentcalculations. For example, if a column retrieved from the |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | database contains 6 significant digits and dec = 0, the column will be |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | show with decimal places (truncated), however any references |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | to the column in subsequent calculations will use 6 decimal places |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | . For example, if the column is referenced in a formula or set function |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | on, all 6 decimal places will be used. |
| `char=` | This mnemonic applies special character(s) to the column's value. | 'x[]x' | Where x references the literal value to display and **[ ]** defines the relative position of the characters (before or after the value).<br>You need only include the **[ ]** if you want to position characters in front of the value. For example, `char='%'` will place a percent sign after the value. If you want to position the word 'minutes' before a value, enter **char='minutes [ ]'**. If you want to output a value like **BUDGET $123.12 (YTD)**, enter **char='BUDGET [ ] (YTD)'**. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| `suppress=` | This is used to indicate a column should not be displayed. A column would be suppressed if it's only defined for use by subsequent columns, for example, if there is a formula that derives a column using two other columns. In this scenario, the columns referenced in the formula can be suppressed. | `true` | |

**Note**

In the case of the zone type Inf


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | `false`<br>Data Explorer-MultipleSQLs(F1_DE), the system will apply the s | This is the default, meaning that you don't need to indicate `suppress=false` at all to indicate that the field should be shown. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | settings for the column definitions in SQL 1 to all subsequent SQLs. | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| `suppressSearch=` | This is used to indicate a column should not be displayed when the zone is invoked in search mode only. | `true` | |

**Note**

In the case of the zone type InfoDataExplorer-Mu


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | `false`<br>ltipleSQLs(F1-DE), thesystemwillapplythesettingsforthecol | This is the default, meaning that you don't need to indicate `suppressSearch=false` at all to indicate that the field should be shown. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | undefinitionsinSQL1toallsubsequentSQLs. | |
| `suppressExport=` | This is used to indicate a column should not be downloaded when exporting. | `true` | |
| | | No | |

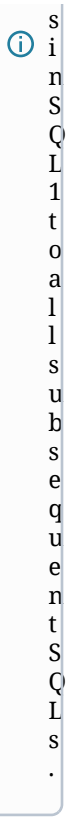

Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | `false`<br>te<br>In the case of the zone type InfoDataExplorer-MultipleSQLs (F | This is the default, meaning that you don't need to indicate `suppressExport=false` at all to indicate that the field should be included in a download. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | 1 – DE), the system will apply the setting for the column definition | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | |  sinSQL1toallsubsequentSQLs. | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `width=` | This is used to override the width of a column (number of pixels). The default value is the maximum width of any cell in the column. | n | Where n is a number between 0 and 999. |

**Note**

If there is no available breaking point in the data, the co


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | lumn will be longer than the specified number of pixels. The length |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | f t h e c o l u m n ' s l a b e l ( w h i c h a p p e a r s i n t h e c o l u m n ' s h e a d i n g ) m a y a l |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | some the width wider than specified. In the case of the zone type In |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | for DataExplorer-MultipleSQLs(F1_DE), the system will set the wi |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |

dthbasedonthecolumndefinitionsinSQL1.


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| `color=` | This is used to override the column's text color. | A valid HTML "named" color | For example `color=yellow`. |

**Note**

Refer to Color Contrast for information about the use of t


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | the HTML color 'red' and its impact on accessibility. |
| | | A valid RGB color model combination | For example `color=#E0292F` or `color=#CCCCCC`. Note that the # is required. |
| `bgcolor=` | This is used to override the column's background color. | A valid HTML "named" color | Similar to the `color=` mnemonic. |
| | | A valid RGB color model combination | Similar to the `color=` mnemonic. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `order=` | Defines the column's default sort order. | ASC | Indicates that the order is ascending. This is the default meaning that it is not necessary to indicate `order=ASC`. |
| | | DESC | Indicates that the order is descending. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `rowHeader=` | Designates the column as a row header for accessibility purposes. | `true` | By default, the first data column of the data explorer results is identified as the row header, for accessibility tools. If the data in the first column does not uniquely identify the row, use this mnemonic to explicitly mark a different column or multiple columns as the row header. |

**Note**

In the case of the zoneTypeInfo DataExplo


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | rer-MultipleSQLs(F1_DE),thesystemwilllookonlyatthecolumn |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | sinSQL1thathavethismnemonicdefinedandapplyittoallresultsf |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | | orthosecolumns. |

### Click Mnemonics

This table describe the mnemonics that define whether a column value may be clicked and if so, what should happen.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `navopt=` | Defines the navigation option that references the target transaction or script when the user clicks a column. Note, this mnemonic should be used in conjunction with the `context=` mnemonic to define what information is sent to the navigation option's target transaction. This mnemonic is ignored if `source=FKREF` because the FK reference code defines the | Cx | This means navigation option code is defined in an earlier column. For example, define `C1` if column 1 defines the navigation option. |
| | | COLUMN_NAME | This means the navigation option was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.<br>Example: `navopt=MAIN_PORTAL` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | hyperlink's destination. | 'Navigation Option Code' | This means the navigation option code is defined directly. For example `navopt='userMaint'`. |
| `context=` | This is used to define one or more context fields and values passed to the target navigation option to go along with the `navopt=` mnemonic. The syntax is as follows: [FIELD_NAME =FIELD_REF FIELD_NAME= FIELD_REF ...] In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the FIELD_NAME, which is the name of the context field in the navigation option. FIELD_REF is the value passed in. The next column indicates the possible values for FIELD_REF. | Cx | Where Cx represents the value of a previous column. For example, if the value to pass is in the first column, reference `C1`. |
| | | COLUMN_NAME | This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. |
| | | 'literal value' | This means a literal value within the single quotes should be passed in. |
| `bpa=` | Indicates that a BPA script should be executed with the user clicks the column | Cx | Indicates that the BPA script is defined in a previous column. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | and indicates the BPA to execute. Note, this mnemonic should be used in conjunction with the `tempstorage=` mnemonic to define the temporary storage values that will be initiated when the script is executed. This mnemonic is ignored if `source=FKREF` because the FK reference code defines the hyperlink's destination. | COLUMN_NAME | This means the BPA script to execute was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. |
| | | 'BPA Script Code' | This means that the BPA script to execute is defined directly. |
| `tempstorage=` | This is used to define how temporary storage variables are initiated when the `bpa=` mnemonic is used. The syntax is as follows: [FIELD_NAME =FIELD_REF FIELD_NAME= FIELD_REF ...] In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the FIELD_NAME, | Cx | Where Cx represents the value of a previous column. For example, if the value to pass is in the first column, reference `C1`. |
| | | COLUMN_NAME | This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | which is the name of the field in temporary storage. FIELD_REF is the value passed in. The next column indicates the possible values for FIELD_REF. | 'literal value' | This means a literal value within the single quotes should be passed in. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| `list=` | This is used to enable work list capability for this column.<br>You may optionally populate the `listdesc=` mnemonic to override the text that will be placed in the worklist zone. | `true` | Setting `list=true` will cause the work list icon to appear in the column's header. If a user clicks the column, it will populate all the rows in the output into the work list zone. |

**Note**

In the case of the zone type InfoDataExplorer-M


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | multipleSQLs(F1-DE),theoutputmaybeshowingaunionoftheresul |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | tsoftmultipleSQLstatements.Inthiscase,ifsomeoftheSQLstate |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |

mentsconfigureagivencolumnwithlist=true,butnotall,onlyth


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |

edata in the cells for the statements that configure this mnemonica


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |

reputintotheworklistwhentheuserclickstheicon.Alsonotetha


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | when determining which columns should have the worklist icon when |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | building the zone, the system only looks at the configuration for t |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | | hecolumnsinSQL1. |
| `listdesc=` | This is an optional mnemonic when using the `list=` mnemonic. It can be used to override the text that is placed in the work list zone. | Cx | Where Cx represents the value of a previous column. For example, if the text to use is in the first column, reference `C1`. |
| `listbroadcast=` | Indicates that the broadcast information for the column is also to be made available in the work list zone. This means that the work list can be used to broadcast information to a portal in the same manner as a data explorer. | `true` | Use this setting to turn on the feature. |

### Common Use Cases

This section provides some common configuration that can be copied and pasted for newly created zones.


Copyright © 2026, Oracle and/or its affiliates.

| Use Case | Code Snippet |
| --- | --- |
| Displaying the description of a lookup value that has been retrieved in the SQL statement. The business service checks for an override label. | **source=BS**<br>**bs='F1-GetLookupDescription'**<br>**label=FLD_NAME**<br>**input=[fieldName='FLD_NAME' fieldValue=FLD_NAME]**<br>**output=description** |
| Display a foreign key with its information but suppressing the navigation. (This is useful when you have a foreign key that is additional information in a zone but would cause confusion to enable navigation. To display foreign key reference with its information and enabled for navigation, use the source of FKREF as described above.)<br>The first example assumes that you know the FK Reference. The second example uses a business service that receives the maintenance object as input. | **source=BS**<br>**bs='F1-GetForeignKeyReference'**<br>**input=[input/fkReference=FK_REF_CD input/fkValue1=BUS_OBJ_CD]**<br>**output=output/infoDescription**<br>**label=DESCR**<br>**source=BS**<br>**bs='F1-GetFKReferenceDetails'**<br>**input=[input/maintenanceObject='F1-EXTLKUP' input/pkValue1=BUS_OBJ_CD input/pkValue2=C1]**<br>**output=output/infoDescription**<br>**label=DESCR** |
| Display a foreign key, including a corresponding navigation link, for generic objects that may or may not have a foreign key reference. If the FK reference exists, its information is shown, otherwise not.<br>Standard foreign key services assume all rows have foreign key details. The service script **F1FKInfoOpt** provides the same foreign key details but does not error when foreign key details do not exist. To build the foreign key description and link, the script needs to be used twice; one column gets the navigation details and another gets the description. The latter references the navigation details column and invokes a BPA script to perform the dynamic navigation. | This column gets the navigation details:<br>**source=SS**<br>**ss='F1FKInfoOpt'**<br>**input=[input/fkReference=C9 input/fkValue1=C6 input/fkValue2=C7]**<br>**output=output/navigationOption**<br>**suppress=true**<br>This column constructs the foreign key description and link:<br>**source=SS**<br>**ss='F1FKInfoOpt'**<br>**input=[input/fkReference=C9 input/fkValue1=C6 input/fkValue2=C7]**<br>**output=output/infoDescription**<br>**label=F1_APPSEC_COMP_DESCR**<br>**bpa='F1ApsecNav'**<br>**tempstorage=[navopt=C15 pkField1=C10 pkValue1=C6 pkField2=C11 pkValue2=C7]**<br>**list=true** |

## Zone Action

Most zone types provided by the product allow for one or more Zone Actions to be defined to appear in the zone header. An action can appear as a hyperlink, icon or button. The action can also be provided as an HTML string.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Zone types also include parameters for actions defined at the zone type level using IMPLEMENTOR_ACTION_n (Action n) parameters. These are rarely used by the product zone types. The actions defined here override any actions defined on the zone type (if present). The details below apply to the zone type level actions as well.

A zone action is defined using the following mnemonics:

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `type=` | This mnemonic defines the appearance of the action in the zone header. | **LINK** | Indicates that the action is shown as a textual hyperlink. |
| | | **ICON** | Indicates that the action is shown as a graphical icon. |
| | | **BUTTON** | Indicates that the action is shown as an HTML button. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | **ASIS** | Indicates that the parameter will provide the HTML to be used for the action. |
| `action=` | This mnemonic defines the action to take when the link/icon/button is clicked. This is ignored when the type=ASIS. | **NAVIGATION** | Indicates that the action is navigation to a page. |
| | | **SCRIPT** | Indicates that the action is to run a BPA script. |
| `navopt=` | Defines the navigation option to use when the `action=NAVIGATION`. | 'NAV_OPT_CD' | Enter a reference to a valid navigation option in single quotes. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| bpa= | Defines the script to run when the action=SCRIPT. | 'SCRIPT_CD' | Enter a reference to a valid BPA script in single quotes. |
| icon= | Indicates the icon to use when type=ICON. | DISP_ICON_CD | Enter a reference to a valid display icon. |
| | | 'path' | Enter an explicit path to the icon, for example 'images/gotoZone.gif'. |
| asis= | This is required when the type=ASIS. This provides the ability to precisely define the HTML you wish to have included in the header. All valid HTML is permitted including the use of "ora" css classes and JavaScript functions. | ['HTML'] | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| label= | By default, the label or tooltip will come from the navigation option or BPA script description. Use this mnemonic to override that label. | FIELD_NAME | Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed. |
| | | 'text' | Enter the text directly in single quotes. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `context=`[target1=source1 target2=source2] | This is used to pass context data when navigating to a page or executing a BPA script. The mnemonic supports passing multiple values.<br>In each case the target context field or BPA script variable is defined first followed by an equal sign, followed by source data defined using one of the valid values defined in the next column.<br>One or more values may be defined. Each context value is defined separated by spaces. The whole set of context values should be surrounded by square brackets. | FIELD_NAME | Indicates that the value should be taken from the field with this name from portal context, global context or the page data model. The mnemonic sourceLoc is used for defining the source. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | xpath | Indicates that the value should be taken from a schema field, represented by the Xpath, displayed in this zone. This is valid when the zone is displaying a UI Map. |
| | | 'constant' | Indicates that the value defined in single quotes should be passed. |
| sourceLoc= | This mnemonic defines the source of the FIELD_NAME's value in the context mnemonic.<br>If this mnemonic is left blank, the default behavior is as follows:<br>- The portal context is checked. | G | Indicates that the field's value is retrieved from the global context. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | - If no portal context value is found, the global context is checked.<br>- If neither value is available, the field is ignored. | P | Indicates that the field's value is retrieved from the portal context. |
| | | D | Indicates that the field's value is retrieved from the page data model. |
| class= | Use this mnemonic to override the look and feel of the link / icon / button using a different CSS style. | 'className1' 'className2' | Enter one or more classes in single quotes. Multiple class names may be provided. |
| style= | Use this mnemonic to override the look and feel of the action element using the indicated css style. | Standard style= format. | All allowed css style definitions may be used. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| vss= | This mnemonic, along with **vinput**= and **voutput**=, may be used to conditionally hide the action. Use this mnemonic to define the service script responsible for determining wherher the action should be visible or not. | 'SCRIPT_CD' | Enter a reference to a valid service script in single quotes. |
| vinput=[target1=source1 target2=source2] | This is used to pass input data to the visibility script. The mnemonic supports passing multiple values.<br>In each case the target field is defined first followed by an equal sign, followed by source data defined using one of the valid values defined in the next column. | FIELD_NAME | Indicates that the value should be taken from the field with this name from portal context. |
| | One or more input values may be defined. Each value is defined separated by spaces. The whole set of values should be surrounded by square brackets. | 'constant' | Indicates that the value defined in single quotes should be passed. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `voutput=` | Use this mnemonic to define the XPath name of a Boolean element set by the visibility script indicating whether to show the action or not. | xpath | If the visibility script sets the element to true the action is visible. If it is set to false the action is not visible. |

Examples:

- **type=BUTTON action=SCRIPT bpa='F1-SET-USER' context=[USER_ID=USER_ID] label=UPDATE_LBL**
- **type=LINK action=NAVIGATION navopt='gotoUser' context=[USER_ID=path(schema/userdId)]**
- **type=LINK … vss='F1BtrActVis' vinput=[batchControl=BATCH_CD batchNumber=BATCH_NBR batchRerunNumber=BATCH_RERUN_NBR batchRunAction='CLOSE_PARMS'] voutput=showAction**
- **type=ASIS asis=['<A class="oraLink" href="www.google.com">Search</a>']**

> **Note**
>
> If the zone type has actions defined and there is a desire to simply remove the zone type actions, the Zone Action can be set with the following configuration: **type=ASIS asis=[]**

## User Filters

Data explorer zones include the ability to define User filters to allow a user to enter data to restrict the zone's rows and / or columns. The filters may be defined individually using User Filter parameters 1–25. Alternatively, a UI map may be defined for capturing filters. In this case, the map's input fields must be associated with the zone's filters by specifying the **xpath=** mnemonic on the respective User Filter parameters.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (`F1-DE`)


Copyright © 2026, Oracle and/or its affiliates.

- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`)
- Info Data Explorer - Single SQL (`F1-DE-SINGLE`)

A user filter is defined using the following mnemonics:

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `name=` | This mnemonic is used if the zone's filter should be pre-populated with a value from global context, portal context or broadcast from another zone. | MD Field Name | |
| `datasource=` | This mnemonic defines the source of the filter's pre-populated value defined in the name mnemonic.<br>If this mnemonic is left blank, the default behavior is as follows:<br>- If the field has been broadcast from another zone, the broadcast value is used.<br>- If no value is broadcast, the portal context is checked to determine if this field exists (if so, its value is taken).<br>- If still no value, the global context is checked.<br>- If still no value, no default value is shown. | `G` | Indicates that the zone should look for the filter value in global context. |
| | | `P` | Indicates that the zone should look for the filter value in portal context. |
| | | `D` | Indicates that the zone should look for the filter value in the page data model. |
| `type=` | Defines the visual metaphor used to capture the filter values. | `DATE` | Filters of this type capture a date. |
| | | `DATE/TIME` | Filters of this type capture a date and time. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | STRING | Filters of this type capture a string |
| | | MONEY | Filters of this type capture a monetary field. This type of filter must also reference the cur mnemonic. |
| | | NUMBER | Filters of this type capture a numeric field. This type of filter may also reference the decimals mnemonic. |
| | | LOOKUP | Filters of this type capture a lookup value. This type of filter must also reference the lookup mnemonic. |
| | | TABLE | Filters of this type capture an administrative table's value (code and description). This type of filter must also reference the table mnemonic. |
| | | CHARTYPE | Filters of this type capture predefined characteristic values for a characteristic type (code and description). This type of filter must also reference the chartype mnemonic. |
| | | ASIS | Filters of this type capture a list of values to be referenced within an 'IN' clause within the SQL statement. |
| label= | Defines the filter's label that appears in the zone's description bar and in the input area. | MD Field Name | Enter a valid field name whose label should be used for the filter label. This should always be the option used if multiple languages are needed. |
| | | 'text' | Defines the text directly. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| cur= | Defines the currency code applied when type=MONEY. | Currency Code | Enter a reference to a valid currency code. |
| dec= | Defines the number of decimal places when type=NUMBER. | Valid number | It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record. |
| lookup= | Defines the lookup flag whose values appear when type=LOOKUP. | Lookup Field Name | Enter a reference to a valid lookup field name. |
| table= | Defines the admin table whose values appear when type=TABLE. | Table Name | Enter a reference to a valid control table name. |
| chartype= | Defines the characteristic type code whose values appear when type=CHARTYPE. | Char Type code | Enter a reference to a valid characteristic type code. |
| xpath= | This mnemonic is used in conjunction with a Filter Area UI Map. For each filter, you must specify the XPath to the corresponding UI map schema element. | XPath | The type= mnemonic must also be appropriate for the map's input field, otherwise the query's SQL could fail. |
| likeable= | This mnemonic defines if a likeable search is performed on the entered value when type=STRING. | S | The query will add % to the suffix of the filter value. |
| | | P | The query will add % to the prefix of the filter value. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | PS | The query will add % to the prefix and suffix of the filter value. |
| `divide=` | The mnemonic controls if a divider line appears above and/or below the filter.<br>Note, you can specify this parameter twice if you want divider lines placed above and below a filter, e.g., **divide=above divide=below**. | `above` | This results in a divider line placed above the filter. |
| | | `below` | This results in a divider line placed below the filter. |
| `searchField=` | This mnemonic controls the initial population of the filter when the zone is launched as a search from a UI map. | MD Field Name | Enter the field name that exactly matches the searchField name specified in the oraSearchField HTML element in the UI map. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| `oraSearchFilter` | This mnemonic is only applicable when the zone is used as a foreign key search. It indicates that this filter is the one that the user is prompted for in the auto-filtering functionality. No additional values are needed. | | |

**Note**

Be sure that a filter that includes this m


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | nemonicalsodefinea'label='mnemonicsothatttheautto-filteringp | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | rompt shows the label. If no label is defined, the user sees the generer | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | iterm'Filter'. | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `encrypt=` | This mnemonic defines if the user filter is encrypted and needs to be searched by hashed value. | [TBL_NAME,FLD_NAME,WHERE_FLD,WHERE_VALUE] | A valid table name and field name are required.<br>The WHERE_FLD and WHERE_VALUE are optional, but if entered, both are required. Use this to only encrypt the field if another field has a certain value. The following is an example.<br>`encrypt=[CI_PERSON,PER_ID_NBR,ID_TYPE_NBR,'SSN']`<br>. The WHERE_VALUE may also reference another filter. The following is an example.<br>`encrypt=[CI_PERSON,PER_ID_NBR,ID_TYPE_NBR,F1].`<br>Note: The field name referenced here should be the source value |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | | | e of the field. However, the SQL should use the hashed value in its filt |

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | | er. |

Examples:

- `label=F1_NBR_DAYS type=NUMBER`
- `label=F1_SHOW_ALL_REQ_FLG type=LOOKUP lookup=F1_SHOW_ALL_REQ_FLG`
- Filter value where a Filter UI Map is defined and Description is one of the filters. `type=STRING xpath=description likeable=S`
  - `type=STRING label=DESCR likeable=S divide=below`
  - `label=REQ_TYPE_CD type=TABLE table=F1_REQ_TYPE`

## Hidden Filters

Data explorer zones include the ability to define Hidden filters to restrict the rows and / or columns that appear in the zone. The following are the potential sources of a hidden filter's value:

- The global area contains the fields whose values are maintained in global context.
- The portal area contains the fields describing the object currently displayed in a portal.
- Other zones on a portal can broadcast information to the portal area, which can then in turn be used by the zone as a hidden filter.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (`F1-DE`)
- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`)
- Info Data Explorer - Single SQL (`F1-DE-SINGLE`)

A hidden filter is defined using the following mnemonics:

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| `name=` | This mnemonic defines the name of the field that needs to be broadcast from other zones or populated in the portal context | FIELD_NAME | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| datasource= | This mnemonic defines the source of the hidden filter's value. If this mnemonic is left blank, the default behavior is as follows:<br>- If the field has been broadcast from another zone, the broadcast value is used.<br>- If no value is broadcast, the portal context is checked to determine if this field exists (if so, its value is taken).<br>- If still no value, the global context is checked.<br>- If still no value, the zone appears as per the poprule mnemonic. | G | Indicates that the zone should look for the filter value in global context. |
| | | P | Indicates that the zone should look for the filter value in portal context. |
| | | D | Indicates that the zone should look for the filter value in the page data model. |
| poprule= | This mnemonic controls what happens if the hidden filter is not present. | R | Indicates that a value for the filter is required. The zone will be set to the "empty state" and the "please broadcast" message will appear in the zone. This is the default value. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | 0 | Indicates that the value is optional. If no value is required, the zone is still built without that value. |
| type= | Defines the visual metaphor used to capture the filter values. | **DATE** | Filters of this type capture a date. |
| | | DATE/TIME | Filters of this type capture a date and time. |
| | | **STRING** | Filters of this type capture a string |
| | | **MONEY** | Filters of this type capture a monetary field. This type of filter must also reference the cur mnemonic. |
| | | **NUMBER** | Filters of this type capture a numeric field. This type of filter may also reference the decimals mnemonic. |
| | | **LOOKUP** | Filters of this type capture a lookup value. This type of filter must also reference the lookup mnemonic. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | | **TABLE** | Filters of this type capture an administrative table's value (code and description). This type of filter must also reference the table mnemonic. |
| | | **CHARTYPE** | Filters of this type capture predefined characteristic values for a characteristic type (code and description). This type of filter must also reference the chartype mnemonic. |
| | | **ASIS** | Filters of this type capture a list of values to be referenced within an 'IN' clause within the SQL statement. |
| `label=` | Defines the filter's label that appears in the zone's description bar. | FIELD_NAME | Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed. |
| | | 'text' | Defines the text directly. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| cur= | Defines the currency code applied when type=MONEY. | CURRENCY_CD | Enter a reference to a valid currency code. |
| dec= | Defines the number of decimal places when type=NUMBER. | N | It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record. |
| lookup= | Defines the lookup flag whose values appear when type=LOOKUP. | LOOKUP_FIELD_NAME | Enter a reference to a valid lookup field name. |
| table= | Defines the admin table whose values appear when type=TABLE. | TABLE_NAME | Enter a reference to a valid admin table name. |
| chartype= | Defines the characteristic type code whose values appear when type=CHARTYPE. | CHAR_TYPE_CD | Enter a reference to a valid characteristic type code. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| searchField= | This mnemonic controls the initial population of the filter when the zone is launched as a search from a UI map. | FIELD_NAME | Enter the field name that exactly matches the searchField name specified in the oraSearchField html element in the UI map. |

## Multi-Select Action

This parameter defines an action to be included in the action area for multi-selection processing. Note that a multi-selection action can only be used if the Multi Select parameter has been set to YES, which causes a checkbox to appear on each row displayed. The action defined here will trigger against all rows selected by the user via the checkbox.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (F1–DE)
- Query Data Explorer - Multiple SQLs (F1–DE-QUERY)
- Info Data Explorer - Single SQL (F1–DE-SINGLE)

A multi select action has the following mnemonics:

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| script= | This mnemonic defines the script to be invoked when the action is clicked. This is required. | SCR_CD | Enter a reference to a valid BPA script or Service Script. |
| type= | This mnemonic | **BUTTON** | The action is rendered as a button. This is the default. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| | defines how the action should be rendered. | **LINK** | The action is rendered as hypertext. |
| | | **ICON** | The action is rendered as a graphic icon. For this option, the icon mnemonic is required. |
| `icon=` | This mnemonic defines the icon to display when `type=ICON`. | DISPLAY_ICON_CD | Enter a reference to a valid display icon code. |
| `refresh=` | This mnemonic indicates how and if a refresh should occur after the script completes. | **NO** | Indicates that no refresh is performed. This is the default. |
| | | **ZONE** | Indicates that a refresh of the zone is performed. |
| | | **PORTAL** | Indicates that a refresh of the entire portal is performed. |
| `label=` | By default, the button label, link text or icon toolti | FIELD_NAME | Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | p will come from the script description. Use this mnemonic to override that label. | 'text' | Enter the text directly in single quotes. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| list= | When executing the script, the framework builds an XML list containing information from each row selected. This list must be defined in the script's schema and referenced in this mnemonic. | listElementName | Enter a valid list element name from the script schema. |
| context=[elementName1=rowData1 elementName2=rowData2] | This mnemonic is used to populate the list with the appr | Cx | Indicates that the element should be populated with a value in the referenced column parameter. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | opriate information from each selected row. The mnemonic supports passing multiple values. In each case the element in the schema list is defined first followed by an equal sign, followed by information about the data used to populate the element defin | Px | Indicates that the element should be populated with a value in the referenced post processing parameter. |
| | | COLUMN_NAME | Indicates that the element should be populated with a value from a column in the SQL statement. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | ed using one of the valid values defined in the next column.<br>One or more values may be defined. Each context value is defined separated by spaces. The whole set of context values should be surrounded by square brackets.<br>Example of a | 'constant' | Indicates that the value defined in single quotes should be passed. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | schema:<br><br><schema><br><accountInfo type="list"><br><accountId/><br><name/><br><amount/><br><process/><br></accountInfo><br></schema><br><br>Example of | | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
| --- | --- | --- | --- |
| | list and context mnemonics. `list=accountInfo` **context=[accountId=ACCT_ID name=C2 amount=P3 process='O']** | | |
| `class=` | Use this mnemonic to override the look and feel of the action using a different CSS style. | 'className1' 'className2' | Enter one or more classes in single quotes to be appended to the standard class(es). Multiple class names may be provided. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Valid Values | Comments |
|---|---|---|---|
| style= | Use this mnemonic to override the look and feel of the action element using the indicated css style. | Standard style= format. | All allowed css style definitions may be used. |

## Data Explorer Graph Configuration

For some data explorer zones, it is more effective to display the results in a graphical form such as a pie or line chart. Visualization option parameters enable the graph configuration to be integrated with the data explorer zone that retrieves the data.

The visualization option parameters support three options:

- **Grid** - displays the data in a standard data explorer grid
- **Chart** - displays the data in the form of a chart
- **Map** - provides the ability to invoke a UI Map to display the data

Up to five visualization option parameters may be specified. One of the options must be the standard grid, allowing for a maximum of four graphical options. The zone will display icons corresponding to the visualization types to allow you to switch between the various views. These icons are presented in the same order are they are defined in the parameter list.

> **Note**
>
> If the only display option is a standard grid, it is not necessary to specify the grid as a visualization parameter.

Visualization option parameters are provided on the Info Data Explorer - Single SQL (`F1-DE-SINGLE`) and Info Data Explorer - Multiple SQLs (`F1-DE`) zone types.

### Visualization Option Parameter Configuration

Mnemonics are used to define the attributes for an option type within the parameter value.


Copyright © 2026, Oracle and/or its affiliates.

For a **grid**, the mnemonic is ‘optiontype=GRID’. No other mnemonics are required.

For a **map**, the mnemonics are ‘optiontype=MAP map=MAP_CD schemaPath=xpath’, where ‘MAP_CD’ is the ID of the UI Map and ‘xpath’ is the schema path of the UI Map. In most use cases the schema elements will match the result set from the data explorer, in which case no schemaPath mnemonic is needed. If the UI Map schema has additional elements present, the schemaPath attribute will define the group node that the data explorer information will be copied into.

For a **chart**, the option type mnemonics is ‘optiontype=CHART’. This is followed by the appropriate chart configuration attributes, which are described in the section below.

### Chart Configuration

The following attributes are used to configure a graphical representation of a set of data.

The ‘`oraChart:`’ attribute defines the type of graph to display and its general configuration. The set of configuration parameters available for this attribute are as follows:

| Parameter | Values | Description |
|---|---|---|
| `type:` | `pie`<br>`stacked`<br>`cluster`<br>`line`<br>`area`<br>`combo` | Defines the type of chart to display.<br>Required |
| `showLegend:` | `true`<br>`false` | Defines if the chart should have a legend displayed.<br>Optional (default is **true**) |
| `legendPosition:` | `left`<br>`right`<br>`bottom`<br>`top` | Defines where the legend should appear.<br>Optional (default is **right**)<br>Setting position to **left** or **right** will automatically render it vertically.<br>Setting position to **top** or **bottom** will automatically render it horizontally. |
| `legendBorder:` | `true`<br>`false` | Defines if the legend should display with a border around it.<br>Optional (default is **false** ) |
| `depth:` | `true`<br>`false` | A value of **true** indicates a 3 dimensional depth for the chart.<br>Optional (default **false**, which is a 2 dimensional chart) |
| `animate:` | `true`<br>`false` | A value of **true** indicates that the graph should animate when displayed.<br>Optional (default is **true**). When using large data sets, consider disabling animation. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
|---|---|---|
| `dataCursor:` | `on`<br>`off` | A value of **on** enables hovering anywhere in the graph.<br>Optional (default is **off**). It is not applicable to pie charts. |
| `orientation:` | `horizontal` | Defines the chart orientation. This only applies to bar, line, area, combo charts.<br>E.g., `oraChart="type:cluster;orientation:horizontal"`, defines horizontal cluster chart.<br>Optional (default is vertical). |

The **oraChartSeries** attribute defines the source information for the graph. Most charts support 5 of these attributes, with the following mnemonics:

- `oraChartSeries1`
- `oraChartSeries2`
- `oraChartSeries3`
- `oraChartSeries4`
- `oraChartSeries5`

Stacked charts support an unlimited number of series by continuing to add attributes `oraChartSeries6` and above, but beware of performance implications and memory limits when using an excessively high number of series. All attributes are identical in format and accept the same parameters, as described below.

> **Note**
>
> All the charts require `oraChartSeries1`. Stacked, Cluster and Line charts may optionally include the additional series attributes (for multiple bars/lines).

If you define multiple series, then data must be provided for all series defined. The data amounts can be 0 (zero) but they have to be present in order for the chart to display correctly.

The set of configuration parameters available for the **oraChartSeriesN** attribute are:

| Parameter | Values | Description |
|---|---|---|
| `list:` | XPath value | Defines the XPath to the list in the XML that contains the data to chart.<br>Required only for HTML definitions in UI Maps. Data explorer values are derived from the columns for the zone. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
|---|---|---|
| `amount:` | Cn<br>Pn<br>XPath value | Defines the element containing the amount to chart. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>Required |
| `xaxis:` | Cn<br>Pn<br>XPath value | Defines the element containing the x-axis data. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element is specified.<br>Required for Stacked, Cluster, Line, Area and Combo charts. |
| `xaxisFormat:` | `date`<br>`dateTime`<br>`time`<br>`localDate`<br>`string` | Defines the x-axis data format.<br>If it is **date**, **dateTime** or **time** then the value is presented in the format as defined on the user's display profile.<br>In case of **localDate** or **string** the data is displayed as is with no special formatting.<br>Optional (Default is **date**). |
| `label:` | Text value | Defines the label for the amount being charted.<br>Either this setting or `labelPath:` must be defined. |
| `labelPath:` | Cn<br>Pn<br>XPath value | Defines the element that provides the label for the amount being charted. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>Either this setting or `label:` must be defined. |
| `currency:` | A valid Currency code | Defines the currency code for the amount being charted.<br>Optional. |
| `currencyPath:` | Cn<br>Pn<br>XPath value | Defines the element that provides the currency code for the amount being charted. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>Optional. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
| --- | --- | --- |
| `hoverText:` | Text value | Defines the hover text for the chart element.<br>Optional (A default hover text is always available.) Ignored if `hoverTextPath:` is defined.<br>The following substitution variables are available.<br>• `$label` This will be replaced with the label text as determined by the `label:` or `labelPath:` setting.<br>• `$amount` This will be replaced with the amount text as specified by the `amount:` setting.<br>• `$axis` This will be replaced with the x-axis text.<br>• $% This will be replaced by the percentage "slice" of the pie or bar.<br>• `$newline` This will force a line break.<br>If the hover text defined contains any of the above values, they will be replaced by the equivalent text prior to being displayed.<br>Example:<br>`"hoverText:$label$newline$amount"` |
| `hoverTextPath:` | Cn<br>Pn<br>XPath value | Defines the element that provides the hover text for the chart element. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>The hover text in the element can utilize all the substitution functionality described above in the `hoverText:` description.<br>Optional. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
| --- | --- | --- |
| `type:` | `bar`<br>`line`<br>`area` | This attribute is used for the **combo** chart type only. It defines how each series on the **combo** chart should be presented.<br>The following example defines a **combo** chart where one series is rendered as bars and another one as area.<br><br>`oraChart="type:combo;" oraChartSeries1="xaxis:C3; label:C2; amount:C1;type:bar" oraChartSeries2="xaxis:C3; label:C2; amount:C1;type:area"` |
| `navOpt:` | A valid Navigation Option code. | Defines a navigation option to be activated when the chart element is clicked.<br>Optional. |
| `navOptPath:` | Cn<br>Pn<br>XPath value | Defines the element that provides the navigation option to be activated when the chart element is clicked. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>Optional<br>Note that both `navOpt:` and `navOptPath:` may be configured. The `navOptPath:` navigation option is processed first. If a value is found it is used, otherwise the value in the `navOpt:` setting is used. This means that the chart definition can include a default navigation option and a navigation option present in the data will override it. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
| --- | --- | --- |
| `key:` | Cn<br>Pn<br>XPath value | Defines the element that contains the key field data to be used in a navigation option. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>This is required if either `navOpt:` or `navOptPath:` is defined.<br>**Note**<br>Only one key field can be configured for a navigation option. |
| `script:` | A BPA script name | Defines a BPA script to be activated when the chart element is clicked.<br>Optional<br>When a script is executed, all the elements from the chart data will be made available to the script as temporary variables. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
| --- | --- | --- |
| `scriptPath:` | Cn<br>Pn<br>XPath value | Defines the element that provides the BPA script to be activated when the chart element is clicked. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>Optional<br>Note that both `script:` and `scriptPath:` can be configured. The `scriptPath:` script option is processed first. If a value is found it is used, otherwise the value in the `script:` setting is used. This means the chart definition can include a default script and a script present in the data will override it. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
| --- | --- | --- |
| `color:` | HTML Color code / RGB value | Optional (default colors applied)<br>Defines the color for the series. The format is a valid HTML color code, e.g. **green** or **blue**. All valid color names are defined in this link: http://www.w3schools.com/htmL/html_colornames.asp.<br>Alternatively an RGB format may be used. (`00FF00` is green and `0000FF` is blue)<br>**Note**<br>Refer to Color Contrast for information about the use of the HTML color 'red' and its impact on accessibility. |
| `colorPath:` | Cn<br>Pn<br>XPath value | Defines the element containing a color for the series. For data explorers, this will be a SQL or post-processing column identifier. For HTML definitions, the XPath of the element must be specified.<br>The valid formats as described in the `color:` setting apply.<br>Optional (default colors applied) |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Values | Description |
|---|---|---|
| `pieColors:` | HTML color code / RGB values | Defines the colors for the pie series. Any number of HTML color codes or RGB color values can be specified, separated by spaces. Examples:<br><br>`pieColors: green blue`<br><br>`pieColors: 00FF00 0000FF`<br><br>Optional (default colors are applied if the series exceeds the values specified) |

The `oraChartBroadcast=`attribute provides the ability to broadcast a value from a chart to the portal context. An unlimited number of fields can be broadcast, configured as name/value pairs; Broadcast is triggered by clicking on a corresponding section of the chart.

The syntax is as follows where FIELD_NAMEn is the name of the portal context field to be broadcast and Cn (or Pn) is the element that contains the data value to be broadcast.

```
oraChartBroadcast="FIELD_NAME1:C1;FIELD_NAME2:C2;"
```

For HTML definitions in UI Maps, ‘xpathn’ is the element in the list schema that contains the data value to be broadcast.

```
oraChartBroadcast="FIELD_NAME1:xpath1;FIELD_NAME2:xpath2;"
```

The `oraChartViewDetails=`attribute provides the ability to launch a view details drawer for an entry in a chart. The syntax is similar to launching a drawer from a UI map or a data explorer column but without rendering an icon. Refer to Launch a View Details Drawer for more information.

UI map examples:

```
oraChartViewDetails="detailPanelName:VIEWDET; USERID:x[user];"
oraChartViewDetails="detailPanelName:x[panelName]; USERID:x[user];"
```

Data explorer examples:

```
oraChartViewDetails="detailPanelName:VIEWDET; USERID:C1;"
oraChartViewDetails="detailPanelName:C2; USERID:C1;"
```

### Zone Configuration Examples

#### Zone with a simple pie chart

In this example, the zone SQL summarizes To Do entries by Role. The column definitions are:

- C1 - role description


Copyright © 2026, Oracle and/or its affiliates.

- C2 - count of To Do entries by role
- C3 - role ID

**Visualization Option 1**

This option shows To Do entries by Role as a pie chart. The role ID will be broadcast to the portal when the corresponding section of the pie chart is clicked.

```
optiontype=CHART
oraChart="type:pie;"
oraChartSeries1="labelPath:C1; amount:C2;hoverText:$label$newline$amount;"
oraChartBroadcast="ROLE_ID:C3"
```

**Zone with two chart types**

In this example, the zone SQL summarizes To Do entries by Priority and also calculates separate counts by To Do status within priority. The column definitions are:

- C1 - priority description
- C2 - count of To Do entries by priority
- C3 - priority flag value
- C4 - count of Open To Dos by priority
- C5 - count of Assigned To Dos by priority
- C6 - Label for the Open To Dos bar in the cluster chart
- C7 - Label for the Assigned To Dos bar in the cluster chart

**Visualization Option 1**

This option shows To Do entries by Priority as a pie chart. The priority value will be broadcast to the portal when the corresponding section of the pie chart is clicked.

```
optiontype=CHART
oraChart="type:pie; legendBorder:true;"
oraChartSeries1="labelPath:C1; amount:C2;hoverText:$label$newline$amount;"
oraChartBroadcast="TD_PRIORITY_FLG:C3"
```

**Visualization Option 2**

This option shows To Do entries by Priority as a cluster chart with two data series, one for Open To Dos and another for Assigned To Dos . The corresponding priority value will be broadcast to the portal when clicking either the Open or Assigned bar of the cluster chart.

```
optiontype=CHART oraChart="type:cluster;  showLegend:true;"
oraChartSeries1="xaxis:C1; xaxisFormat:string; labelPath:C6;
amount:C4;hoverText:$label$newline$amount;"
oraChartSeries2="xaxis:C1; xaxisFormat:string; labelPath:C7;
amount:C5;hoverText:$label$newline$amount;"
oraChartBroadcast="TD_PRIORITY_FLG:C3"
```

## Pagination Configuration


Copyright © 2026, Oracle and/or its affiliates.

The various data explorer zones in the product support the ability to configure pagination so that a user can 'page through' a large set of results using "previous" and "next" buttons or links.

There are several zone parameters that are impacted when attempting to configure this functionality. The following steps highlight the configuration.

- The **Enable Pagination** parameter must be configured to define the basic setup for pagination for the zone. This parameter defines whether the "previous" and "next" actions are defined as buttons, links or icons and indicates the location of the actions. It also allows an indication as to whether the additional rows are simply appended, rather than shown in a new "page". Refer to the parameter's embedded help for information about the specific syntax.
- It is recommended that the zone is configured with record count and page information by properly configuring the **Record Count Display** parameter. Refer to the parameter's embedded help for information about the specific syntax.
- Configure the **Number of Rows to Retrieve for SQL**. parameter to define the number of records displayed per page. If this parameter is not specified the value in the **Number of Rows to Display** parameter is used.
- Configure the key that will be used for paging so that the system can keep track of the 'page break'. The data must be sorted by the paging key; as a result, the decision for identifying the paging key must take into account the design for the zone and the data being displayed. In addition, the paging key must be unique to ensure that the page breaks occur correctly. See below for configuration examples.
  - The **SQL Statement** must includes additional clauses **PAGENEXT** and **PAGEPREV** based on the paging key. In addition, as mentioned above, the paging key must be used in the ORDER BY clause.
  - The SQL **Column** parameters must define the paging key mnemonic to be used in conjunction with the SQL statement paging clauses.
  - It is recommended to configure the **SQL Display Column** parameter to show the data in the same order as the ORDER BY clause.

The following zone types support this capability:

- Info Data Explorer - Single SQL (`F1-DE-SINGLE`).
- Info Data Explorer - Multiple SQLs (`F1-DE`). Note that zones of this type support a union of the results of all the SQL statements. As a result, pagination may only be enabled for zones of this type if a single SQL is used. The system is not able to keep track of the pagination across disparate SQL statements.
- Query Data Explorer - Multiple SQLs (`F1-DE-QUERY`).
- Multi Query Data Explorer (`F1-DE-MULQRY`). Zones of this type do not include configuration for SQL statements or column display. However, they do include configuration for the **Enable Pagination**. This parameter must be configured in order for pagination on the individual zones to work.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Zones used for a Business Service.** Note that pagination is ignored when invoking a data explorer zone via a business service. In this scenario, the zone will return the first "chunk" of rows as defined by the Number of Rows parameters.
>
> **Zones with data visualization options.** Pagination is not supported for zones that define visualization options to show data in a graphical form.

### Examples

#### Simple Paging Key

In this example, the Extendable Lookup Value is defined as Column 1 (C1) and is marked as the paging key. This field is unique for the table and works well as a simple paging key.

```
SELECT A.F1_EXT_LOOKUP_VALUE,A.BUS_OBJ_CD
FROM
  F1_EXT_LOOKUP_VAL A,
  F1_EXT_LOOKUP_VAL_L B
WHERE
A.BUS_OBJ_CD = :H1
AND A.BUS_OBJ_CD = B.BUS_OBJ_CD
AND A.F1_EXT_LOOKUP_VALUE = B.F1_EXT_LOOKUP_VALUE
AND B.LANGUAGE_CD = :LANGUAGE
[(F1) AND UPPER(A.F1_EXT_LOOKUP_VALUE) like UPPER(:F1)]
[(F2) AND ((UPPER(B.DESCR_OVRD) like UPPER(:F2))
OR (B.DESCR_OVRD = ' ' AND UPPER(B.DESCR) like UPPER(:F2)))]
[(PAGENEXT) AND A.F1_EXT_LOOKUP_VALUE > :C1]
[(PAGEPREV) AND A.F1_EXT_LOOKUP_VALUE < :C1]
ORDER BY A.F1_EXT_LOOKUP_VALUE
```

#### Complex Paging Key

Most queries however do not sort by a unique value. In this case, the paging key needs to be set based on the sorting of the query and should include a unique field, such as the primary key, as the last paging key. In this example, the query is showing results sorted by To Do Type, Role and User. All fields, including the To Do Entry ID (the primary key) are marked as paging keys.

```
SELECT TD_TYPE_CD, ROLE_ID, ASSIGNED_TO, ASSIGNED_DTTM, TD_PRIORITY_FLG,
TD_ENTRY_ID
FROM  CI_TD_ENTRY
WHERE
ENTRY_STATUS_FLG IN ('O', 'W')
[(F1) and TD_TYPE_CD = :F1]
[(F2) AND ASSIGNED_TO = :F2]
[(F3) AND ROLE_ID = :F3]
[(PAGENEXT) and ((TD_TYPE_CD>:C1) or (TD_TYPE_CD=:C1 and ROLE_ID>:C2) or
(TD_TYPE_CD=:C1 and ROLE_ID=:C2
and ASSIGNED_TO>:C3) or (TD_TYPE_CD=:C1 and ROLE_ID=:C2 and ASSIGNED_TO=:C3
AND TD_ENTRY_ID>:C4))]
[(PAGEPREV) and ((TD_TYPE_CD<:C1) or (TD_TYPE_CD=:C1 and ROLE_ID<:C2) or
(TD_TYPE_CD=:C1 and ROLE_ID=:C2
```


Copyright © 2026, Oracle and/or its affiliates.

```
and ASSIGNED_TO<:C3) or (TD_TYPE_CD=:C1 and ROLE_ID=:C2 and ASSIGNED_TO=:C3
AND TD_ENTRY_ID<:C4))]
ORDER BY TD_TYPE_CD, ROLE_ID, ASSIGNED_TO, TD_ENTRY_ID
```

## Auto-filtering Configuration

Auto-filtering is the term used to describe functionality available in search fields where a user can type search text without having to open the search. Refer to Input Field with Search for examples of this feature. This section describes how to enable this feature and how to configure a search zone to enable this functionality.

The feature is opt-in. To turn on the feature, navigate to Feature Configuration. Search for an existing feature configuration with the Feature Type `Custom Look and Feel`. If one does not exist, create a feature configuration of this type. Select the `Enable Auto-Filtering` option and set the value to 'Y'.

The feature supports designating a single user filter in the zone as the one to expose to the user for auto-filtering. For example, when searching for a control table record, like To Do Type, the system configures the Description filter as the one that supports auto-filtering. For an object like Person, the person name filter may be the one that is configured for this feature.

To indicate the filter to support auto-filtering, add the mnemonic `oraSearchFilter` to the filter definition. For example, here is the configuration for the description user filter in the To Do Type search zone.

```
type=STRING label=DESCR likeable=S xpath=descr oraSearchFilter
```

In addition, when turning on this feature, the system supports displaying up to 3 columns of content for each record to help the user choose the correct value. The best practice is ensure that at least one of the columns configured to display matches the filter the user is typing. For example, the system has configured that all control table searches show both the code and the description. This configuration goes into the `Initial Display Columns` parameter, or for Query data explorers, if individual SQLs have different display columns, add the configuration to the appropriate `SQL n Display Column` parameter. The syntax is the mnemonic `oraSearchDisplay=` followed by a comma delimited list of up to 3 column references in the order to display. Note that the columns in the search display mnemonic are limited to those referenced in the display parameter. Example:

```
C1 C2 C3 C4 C5 oraSearchDisplay='C2,C1'
```

## Use Data Explorer for Derived Data

There are times when a design warrants displaying data in a data explorer zone that is not accessible via SQL. For example, perhaps the data is from another system and it requires a web service call. The JMS Message Browser is another example.

The product provides functionality in the data explorer that allows you to call a script after the user filters are populated but before the SQL is executed. The script can retrieve the data as appropriate, store the data in table format so that the SQL can retrieve the data from the table.

The following points provide more detail:

- Create a service script that retrieves the data as needed. This script should store the retrieved data in a temporary table.


Copyright © 2026, Oracle and/or its affiliates.

  - The product provides a table that may be used. It is called F1_GENERIC_GTT (Generic Global Temporary Table). There is a business service — Create Global Temporary Table Records (`F1-InsertGTTRecords`) that the service script may call to insert the records.
  - Note that if the data is accessed via a web service call, it may be appropriate to execute the web service in a separate session using the business service F1-ExecuteScriptInNewSession to trap errors that may be issued by the web service call and provide a better error.
- In the data explorer zone use this service script in the zone's pre-processing script parameter. If any user or hidden filters should be passed into the script, the parameter supports mnemonics for this purpose. Refer to the parameter's embedded help for the supported syntax.
- The SQL for the data explorer should access the temporary table that was populated by the service script.

## Configuring Timeline Zones

This topic highlights information related to configuring a timeline zone. The zone type is `F1-TIMELINE`. A timeline zone contains one or more "lines" where each line shows when significant events have occurred. The output of each line is driven by an algorithm configured on a timeline zone. Each algorithm is responsible for retrieving a single type of data. For example, one algorithm may retrieve bills for an account in a given time period whereas another algorithm may retrieve payments for that account for the same time period.

The algorithms to configure for a timeline zone use the **Zone - Timeline** plug-in spot. Please note the following details about the behavior for algorithms for this plug-in spot.

- The timeline algorithm receives all the global context values currently populated. In addition, it receives a start and end date from the zone, based on the time period chosen by the user, along with the maximum number of events that can be reasonably displayed for the chosen time period. The algorithm should use this information to retrieve data for a given type of transaction related to one or more of the input context values for the provided time period.
- For each event found, the algorithm returns information about the event along with many options that assist the user in getting more detail about each event or acting on an event.
  - Event date
  - Primary key of the record (key / value pairs)
  - FK Reference. With this information, the timeline zone will display the appropriate info string to display in the zone's info area when clicking on the event. In addition, the FK reference identifies the appropriate navigation option to use when a user clicks the info string hypertext to view the record on its maintenance page.
  - Background Color and Text Color to use for the event. (Obsolete). This information is no longer used by the timeline zone and will be ignored.
  - Icon use for the event. (Obsolete). This information is no longer used by the timeline zone and will be ignored.
  - BPA script to launch when a user clicks on an event. (Optional). The algorithm may return one or more BPA scripts that a user may launch to act on an event. For example, for an event that has a status of **Error**, perhaps a BPA is provided to walk a user through resolving the error.


Copyright © 2026, Oracle and/or its affiliates.

When a script is initiated from a timeline, the system puts the prime key of the event into a field in the page data model. The name of the field is the column name(s) of the event's prime key. For example, when a script associated with a payment event is kicked off, the system populates a field called PAY_ID with the prime-key of the selected payment.

Note that your specific edge application may supply algorithm types for a timeline zone as part of the base product. Although algorithm types may be provided, typically the product does not deliver algorithms because the parameters for the algorithms are driven by a particular implementation's business rules and preferences. As a result, the product will also not deliver pre-configured timeline zones. Please refer to your edge application's documentation for more information about what timeline algorithm types are delivered, if any and recommendations for configuration.

# Defining Context-Sensitive Zones

A context-sensitive zone allows you to associate a zone with a specific user-interface transaction. A context-sensitive zone appears at the top of the Sidebar when a user accesses a page for which the zone is specified as the context.

> ⚠ **Caution**
>
> Make sure that the zone is appropriate for the transaction on which you are specifying it.

Select **Admin**, and then **Context Sensitive Zone** to maintain context-sensitive zones.

**Description of Page**

The **Navigation Key** is a unique identifier of a tab page within the system. **Owner** indicates if this navigation key is owned by the base product or by your implementation (**Customer Modification**).

> ⚠ **Caution**
>
> Important! When introducing a new context sensitive zone, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The grid contains the list of context-sensitive zones and the sequence in which they appear in the sidebar for the selected navigation key. The grid contains the following fields:

- **Zone** is the name of the zone to display in the Sidebar.
- **Sequence** is the sequence in which the zone is displayed (if multiple context-sensitive zones are defined).
- **Owner** indicates if this context sensitive zone is owned by the base product or by your implementation (**Customer Modification**).

**Where Used**

A context-sensitive zone displays at the top of the Sidebar whenever a user accesses the transaction for with the zone is specified.


Copyright © 2026, Oracle and/or its affiliates.

## Defining Portals

This portal is used to view and maintain the configuration of portals. An implementation may define their own portals or may override some of the settings for base product provided portals.

You can access the portal from the **Admin Menu**, and then **System**, and then **Portal**. You are brought to a query portal with options for searching for a specific record. Once a portal record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Understanding Portals and Zones for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Portal.** This zone provides information about the selected portal definition.
- **Tabs.** This zone lists other tab portals if any are configured for a standalone portal. This zone is only visible when such tabs exist for the current standalone portal.
- **Context Sensitive Zones.** This zone lists context sensitive zones that appear on the sidebar when this portal is displayed . This zone is only visible when such zones exist for the current standalone portal.

The following zones may appear as part of the portal's **Options** tab page

- **Options.** This zone lists various configuration options for the selected portal record.

Some portals may be associated with configurable settings that are maintained on a **Configuration** tab. These settings are defined using a master configuration business object referenced as portal option. The tab is visible when such option exists for the portal.

# Defining Menu Options

This portal is used to define / change any menu in the system. You can access the portal from the **Admin Menu**, and then **System**, and then **Menu**. You are brought to a query portal with options for searching for a specific menu. Once a menu has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Caution**
>
> Updating menus requires technical knowledge of the system. This is an implementation and delivery issue and should not be attempted if you do not have previous experience with menus.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Security and menus.** Refer to Secured Objects and Application Services for a discussion of how application security can prevent menu items (or an entire menu) from appearing.

> **Note**
>
> **Module configuration and menus.** Your module configuration can prevent menu items (or an entire menu) from appearing.

The following zones may appear as part of the portal's **Main** tab page:

- **Menu.** This zone provides information about the selected menu.
- **Menu Lines.** This zone lists the menu's lines and supports actions to add, edit and delete lines. You may also broadcast a line to review more information about its items configuration.
- **Menu Line Items.** This zone lists item information for a selected menu line.

Application security information for the menu can be reviewed and managed from the **Application Security** tab. The following zones may appear as part of this tab page:

- **Menu Lines.** This zone lists the menu's lines. Broadcast a line to review its application security information.
- **Menu Line Security.** This zone lists application services related to the selected menu line. You may use this zone to review and configure security access for the selected menu line and a specific user group.
- **Menu Line Security Details.** This zone lists the entities secured by a selected application service.

The following zone may appear as part of the portal's **References** tab page:

- **Menu References.** This zone lists entities that reference the selected menu.

The following sections provides more information.

### Menu Types

The following types of menus are supported:

- **Main** menu type defines the menu items under the Main Menu entry visible when the menu is initially expanded.
- **Admin** menu type defines the menu items under the Admin Menu entry visible when the menu is intially expanded.
- **Submenu** defines a menu group that appears when the main menu or admin menu is selected. For the Admin menu, this is only visible when it's organized functionally. Refer to Admin Menu for more information about options for organizing the admin menu.
- **Context** refers to a context menu.
- **Page Action Menu** defines buttons that appear in the Page Title Area.
- **User Menu** refers to the menu items that appear on the user menu; for example, User Preferences.


Copyright © 2026, Oracle and/or its affiliates.

#### Extending Product Owned Menus

An implementation may choose to add custom menu lines along with its menu item (or items) to a base product owned menu.

#### Overriding Menu Lines Sequence

An implementation may override the sequence of a base product owned menu line. Also note that the sequence is defined on the menu line language table, allowing for different orders to be used for different languages (or to let the menu be sorted alphabetically in one language and in a specified order in a different one).

#### Menu Information is Cached

Use the **Flush Menu** button to flush the cached menu items so you can see any modified or newly created menus. Refer to Caching Overview for more information.

# Defining Navigation Keys

Each location to which a user can navigate (e.g., transactions, tab pages, tab menus, online help links, etc.) is identified by a navigation key. A navigation key is a logical identifier for a URL.

- Navigation Key Types
- Navigation Key vs Navigation Option
- The Flexibility of Navigation Keys
- Linking to External Locations
- Overriding Navigation Keys
- Maintaining Navigation Keys

## Navigation Key Types

There are three types of navigation keys:

- **System navigation keys** define locations where the target for the navigation is a transaction or portal within the system. The navigation keys define the program component that identifies the page to navigate to.
- **External navigation keys** define a URL that identifies the target location. External URLs can be specified as relative to the product web server or fully qualified. External navigation keys always launch in a new instance of a browser window. Examples of external navigation keys include URLs to external systems.
- **Help navigation keys** define a online help topic that identifies the specific page within the online help to launch. Help navigation keys may be related to a program component when the help is related to a specific page in system.

## Navigation Key vs Navigation Option

The system has two entities that work in conjunction with each other to specify how navigation works:

- Navigation Key defines a unique location to which a user can navigate. For example, each page in the system has a unique navigation key. Navigation keys can also define locations


Copyright © 2026, Oracle and/or its affiliates.

that are "outside" of the system. For example, you can create a navigation key that references an external URL. Think of a navigation key as defining "where to go".

- Navigation Option defines how a page is opened when a user wants to navigate someplace. For example, you might have a navigation key that identifies a specific page. This navigation key can then be referenced on two navigation options; the first navigation option may allow users to navigate to the page with no context included, while the second navigates to the page with context data provided to automatically display information related to that context.
- Please note that a wide variety of options can be defined on a navigation option. In addition to defining if data is passed to the page, it could also define search options. In addition, there are some navigation options that do not reference a navigation key but rather refer to a BPA script that should be launched.

## The Flexibility of Navigation Keys

Navigation keys provide a great deal of functionality to your users. Use navigation keys to:

- Allow users to navigate to new pages or search programs
- Allow users to transfer to an external system or web page. After setting up this data, your users may be able to access this external URL from a menu, a context menu, their favorite links, etc. Refer to Linking to External Locations for more information.

Refer to the Tool Suite Guide for more information on developing program components.

> **Note**
>
> **Replacing Base-Package Pages or Searches**. If your new page or search has been designed to replace a module in the base-package, the navigation key must indicate that it is overriding an existing navigation key.

## Linking to External Locations

If you want to include links to external systems or locations from within the system, you need to:

- Define a navigation key that specifies the URL of the location. For example, define an external navigation key that as a URL of **http://www.oracle.com/**.
- Define a navigation option that specifies from where in the system a user can go to your external location. For example, define a navigation option with a usage of **Favorites** or with a usage of **Menu**. Your navigation option points to the navigation key you defined above.
- Add your navigation option to the appropriate location within the system. For example, have users add the navigation option to their Favorite Links or add the navigation option as an item on a menu.

## Overriding Navigation Keys

Your implementation may choose to design a program component (e.g., a maintenance transaction or search page) to replace a component provided by the system. When doing this, the new navigation key must indicate that it is overriding the system navigation key. As a result, any menu entry or navigation options that reference this overridden navigation key automatically navigates to the custom component.


Copyright © 2026, Oracle and/or its affiliates.

For example, if you have a custom On-line Batch Submission page and would like users to use this page rather than the one provided by the system, setting up an override navigation key ensures that if a user chooses to navigate to the On-line Batch Submission from Menu or a context menu, the user is brought to the custom On-line Batch Submission page.

To create an override navigation key, you need to:

- Define a navigation key using an appropriate naming convention.
- If the Navigation Key Type of the navigation key being overridden is **External**, specify a Navigation Key Type of **Override (Other)** and define the appropriate URL Component.
- If the Navigation Key Type of navigation key being overridden is **System**, specify a Navigation Key Type of **Override (System)** and populate the Program Component ID with your custom program component ID.
- Specify the navigation key that you are overriding in the **Overridden Navigation Key** field.

Refer to the Tool Suite Guide for more information about developing your own program components.

# Maintaining Navigation Keys

Select **Admin**, and then **System**, and then **Navigation Key** to maintain navigation keys.

Use the **Navigation Key Query** portal to search for an existing navigation key. Once a navigation key is selected, you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! When introducing a new navigation key, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

### Description of Page

The **Navigation Key** is a unique name of the navigation key for internal use.

**Owner** indicates if this navigation key is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

**Navigation Key Type** includes the following possible values:

- **External** indicates that the location is specified in the **URL Component** field.
- **Help** indicates that the navigation key is used to launch online help where the specific help topic is defined in the **URL Component** field.
- **Override (Other)** indicates that the navigation key overrides another navigation key of type **External** or **Help**. For this option, the name of the navigation key being overridden is populated in the **Overridden Navigation Key** field.
- **Override (System)** indicates that the navigation key overrides a system navigation key. For this option, the name of the navigation key being overridden is populated in the **Overridden Navigation Key** field.
- **System** indicates that the navigation key refers to a transaction in the system identified by its program component.


Copyright © 2026, Oracle and/or its affiliates.

**NOT_SUPPORTED**

Refer to Navigation Key Types for more information about navigation key types.

**NOT_SUPPORTED**

Refer to Overriding Navigation Keys for more information about settings required to override a system navigation key.

**Program Component ID** is the name of the program component identified by the key (for system navigation keys). The program component ID can also be used to specify the transaction with which an online help link is associated.

**Overridden Navigation Key** is the name of the navigation key that the current navigation key is overriding (if **Override (Other)** or **Override (System)** is selected for the **Navigation Key Type**). Refer to Overriding Navigation Keys for more information.

**URL Component** is the specific URL or portion of the URL for the navigation key (external and help navigation keys only). The URL can be relative to the product web server or fully qualified.

**Open Window Options** allows you to specify options (e.g., width and height) for opening a browser window for an external navigation key. (External navigation keys always launch in a new browser window.) You can use any valid features available in the Window.open( ) JavaScript method. The string should be formatted the same way that it would be for the features argument (e.g., `height=600,width=800,resizeable=yes,scrollbars=yes,toolbar=no`). Refer to a JavaScript reference book for a complete list of available features.

**Application Service** is the application service that is used to secure access to transactions associated with **External** navigation keys. If a user has access to the specified application service, the user can navigate to the URL defined on the navigation key. Refer to The Big Picture of Application Security for more information.

The grid displays menu items that reference the navigation key (actually, it shows menu items that reference navigations options that, in turn, reference the navigation key).

# Defining Navigation Options

Every time a user navigates to a transaction, the system retrieves a navigation option to determine which transaction should open. For example:

- A navigation option is associated with every menu item. When a user selects a menu item, the system retrieves the related navigation option to determine which transaction to open.
- A navigation option is associated with every favorite link. When a user selects a favorite link, the system retrieves the related navigation option to determine which transaction to open.
- A navigation option is associated with every node in the various trees. When a user clicks a node in a tree, the system retrieves the related navigation option to determine which transaction to open.
- Etc.


Copyright © 2026, Oracle and/or its affiliates.

Many navigation options are shipped with the base product and cannot be modified as these options support core functionality. As part of your implementation, you may add additional navigation options to support your specific business processes.

Navigation options may also be used to launch a BPA script.

The topics in this section describe how to maintain navigation options.

> ⚠ **Caution**
>
> In order to improve response times, navigation options are cached the first time they are used after a web server is started. If you change a navigation option and you don't want to wait for the cache to rebuild, you must clear the cached information so it will be immediately rebuilt using current information. A special button has been provided on the Main tab of the navigation option transaction that performs this function. Please refer to Caching Overview for information on the various caches.

- Navigation Option - Main
- Navigation Option - Tree

# Navigation Option - Main

Select **Admin**, and then **System**, and then **Navigation Option** to maintain a navigation option.

Use the **Navigation Option Query** portal to search for an existing navigation option. Once a navigation option is selected, you are brought to the maintenance portal to view and maintain the selected record.

**Description of Page**

Enter a unique **Navigation Option** code and **Description**.

> ⚠ **Caution**
>
> When introducing a new navigation option, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The **Flush System Login Info** button is used to flush the cached navigation options so you can use any modified navigation options. Refer to Caching Overview for more information.

**Owner** indicates if this navigation option is owned by the base product or by your implementation (**Customer Modification**). This field is display-only. The system sets the owner to **Customer Modification** when you add a navigation option.

> ⓘ **Note**
>
> You may not change navigation options that are owned by the base product.

Use **Navigation Option Type** to define if the navigation option navigates to a **Transaction**, launches a BPA **Script** or opens an **Attachment**.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The **Attachment** option type is only applicable to navigation options that are used to launch a file attached to a record in the Attachment maintenance object.

For navigation option types of **script**, indicate the **Script** to launch. You can use the **Context Fields** at the bottom of the page if you want to transfer the contents of specific fields to temporary storage variables available to the script. The script engine creates temporary storage variables with names that match the Context Field names.

For navigation option types of **transaction**, define the **Target Transaction** (navigation key) and optionally a specific **Tab Page** (also a navigation key) if a specific tab on the transaction (other than the Main tab) should be opened when navigating to that transaction.

> **Note**
>
> **Finding transaction navigation keys**. When populating the **Target Transaction** and **Tab Page** you are populating an appropriate navigation key. Because the system has a large number of transactions, we recommend using the "%" metaphor when you search for the transaction identifier. For example, if you want to find the currency maintenance transaction, enter "%currency" in the search criteria.

The additional information depends on whether the target transaction is a fixed page or a portal-based page:

- For portal-based pages:
  - **Navigation Mode** is not applicable and should just be set to **Add Mode**.
  - If navigating to a query portal, by default the query portal will open with the default search option defined. If the navigation should open a different search option, define the **Multi-Query Zone** for that query portal and indicate the **Sub-Query Zone** to open by default. Note that for this configuration, it is common to define **Context Fields** to pre-populate search criteria in the target query zone. When using this configuration, be sure that the target query zone's user filters are defined to populate data from context.
- For fixed pages:
  - **Navigation Mode** indicates if the **Target Transaction** should be opened in **Add Mode** or **Change Mode**.
  - **Add Mode** should be used if the option is used to navigate to a transaction ready to add a new object. You can use the **Context Fields** at the bottom of the page if you want to transfer the contents of specific fields to the transaction when it opens.
  - **Change Mode** is only applicable for fixed pages and should be used if the option is used to navigate to a transaction ready to update an object. You have two ways to define the object to be changed:
    - Define the name of the fields that make up the unique identifier of the object in the **Context Fields** (and make sure to turn on **Key Field** for each such field).
    - Define the **Search Transaction** (navigation key) if you want to open a search window to retrieve an object before the target transaction opens. Select the appropriate **Search Type** to define which search method should be used. The options in the drop down correspond with the sections in the search (where **Main** is the first section, **Alternate** is the 2nd section, **Alternate 2** is the 3rd section, etc.).


Copyright © 2026, Oracle and/or its affiliates.

You should execute the search window in order to determine what each section does.

When you select a **Search Type**, define appropriate **Context Fields** so that the system will try to pre-populate the search transaction with these field values when the search first opens. Keep in mind that if a search is populated with field values the search is automatically triggered and, if only one object is found that matches the search criteria, it is selected and the search window closes.

- **Search Group** is only visible if the **Development Tools** module is not turned off. It is used to define the correlation between fields on the search page and the tab page. You can view a tab page's **Search Groups** by viewing the HTML source and scanning for **allFieldPairs**.

The **Go To Tooltip** is used to specify the label associated with the tool tip that appears when hovering over a **Go To** object. Refer to the **Usage** grid below.

The **Usage** grid defines the objects on which this navigation option is used:

- Choose **Favorites** if the navigation option can be used as a favorite link.
- Choose **Menus** if the navigation option can be used as a user's home page or as a menu or context menu item.
- Choose **Script** if the navigation option can be used in a script.
- Choose **Foreign Key** if the navigation option can be used as a foreign key reference.
- Choose **Go To** if the navigation option can be used as a "go to" destination ("go to" destinations are used on Go To buttons, tree nodes, and hyperlinks).
- If your product supports marketing campaigns, you can choose **Campaign** if the navigation option can be used as a "post completion" transaction on a campaign. For more information refer to that product's documentation for campaigns.

The **Context Fields** grid contains the names of the fields whose contents will be passed to the **Target Transaction** or **Script** or used to launch an **Attachment**. The system retrieves the values of these fields from the "current" page and transfers them to the target transaction or to the script's temporary storage. Turn on **Key Field** for each context field that makes up the unique identifier when navigating to a transaction in **Change Mode**.

> **Note**
>
> For an **Attachment**, the grid should contain the Attachment ID.

> **Note**
>
> **Navigating from a Menu**. The standard followed for many base main **menu** navigation options for fixed transactions is that navigation options launched from the Menu dropdown are configured with no context.

# Navigation Option - Tree

This page contains a tree that shows how a navigation option is used. Select **Admin**, and then **System**, and then **Navigation Option** and navigate to the **Tree** tab to view this page.

**Description of Page**


Copyright © 2026, Oracle and/or its affiliates.

The tree shows every menu item, favorite link, and tree node that references the navigation option. This information is provided to make you aware of the ramifications of changing a navigation option.

# Navigation to External Pages

The URL Navigation (F1-URLNavigation) extendable lookup defines named external pages to which a user may navigate to from the application. For example, a named code for the Service Request page in Oracle Fusion.

The actual URL for the page is specific to your implementation and as such is not delivered on the extendable lookup record. When applicable, update the entry to reference the URL for the corresponding page. The UI location that references the name external page shows the corresponding URL only when a URL is provided.

# Understanding System Messages

All error, warning and informational messages that are displayed in the system are maintained on the message table. Every message is identified by a combination of two fields:

- **Message category number.** Think of a message category as a library of messages related to a given functional area. For example, there is a message category for billing messages and another one for payment messages.
- **Message number**. A unique number identifies each message within a category.

Every message has two components: a brief text message and a long description.

The brief message text supports the %n notation within the message text to cause field values to be substituted into a message. For example, the message text "Unable to assign To Do Entry (%1). It is already assigned to another User (%2)" will have the values of 2 fields merged into it before it is displayed to the user (%1 is the To Do Entry ID, %2 is the current assigned user).

> **Note**
>
> **You cannot change the text provided with the product.** If the message is "owned" by the product, you cannot change the product's message or detailed description. If you want your users to see a different message or detailed description other than that supplied by the product, enter your desired verbiage in the corresponding override fields (and flush the cache).

- Defining System Messages

## Defining System Messages

This portal is used to search for message categories and messages that appear throughout the system. An implementation may introduce messages used in custom processes or may choose to override the text for messages delivered by the product.

You can access the portal from the **Admin Menu**, and then **System**, and then **Message**. You are brought to a query portal with options for searching for a specific message category or a message. Once a record has been selected you are brought to the maintenance portal to view and maintain the selected record.


Copyright © 2026, Oracle and/or its affiliates.

Refer to Understanding System Messages for more information.

- Defining Message Categories
- Defining Messages

## Defining Message Categories

This portal is used to view and maintain a message category and its messages.

> **⚠ Caution**
>
> Message category 80000 or greater must be used to define new messages introduced for a specific implementation of the system. Changes to other Message Text will be overwritten when you next upgrade. If you want to make a change to a Message, drill down on the message and populate the Override Message Text. Note that even for message categories 80000 and higher, message numbers lower than 1000 are reserved for common base product messages.

You can access the portal from the **Admin Menu**, and then **System**, and then **Message**. You are brought to a query portal with options for searching for a specific message category or a specific message. Once a message category has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding System Messages for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Message Category.** This zone provides information about the selected message category.
- **Messages.** This zone lists and maintains messages for the selected category.

## Defining Messages

This portal is used to view and maintain a specific message. An implementation may introduce messages used in custom processes or may choose to override the text for messages delivered by the product.

> **⚠ Caution**
>
> Message category 80000 or greater must be used to define new messages introduced for a specific implementation of the system. Changes to other Message Text will be overwritten when you next upgrade. If you want to make a change to a Message, drill down on the message and populate the Override Message Text. Note that even for message categories 80000 and higher, message numbers lower than 1000 are reserved for common base product messages.

You can access the portal from the **Admin Menu**, and then **System**, and then **Message**. You are brought to a query portal with options for searching for a specific message category or a specific message. Once a message has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding System Messages for more information.

The following zones may appear as part of the portal's **Main** tab page


Copyright © 2026, Oracle and/or its affiliates.

- **Message.** This zone provides information about the selected message.

# Understanding Unified Search

Depending on your product, the toolbar search box may support a product specific search for business entities in addition to the standard search for menu items. For example, the search may support a query of business entities by their name, address, contact details etc. The unified search feature is a simplified version of your product's main search, that allows the user to lookup up records using free form search text or hint codes. For more complex queries, the user may navigate to the corresponding advanced search portal.

The configuration of a unified search option includes the following items:

- A **search service script** that implements the data query and defines the supported **filters**. The application service associated with this script is used to secure the unified search option.
- **Search filter rules** and **hints** for the supported filters. These rules are used to parse the text entered by the user to determine whether it qualifies as a valid filter value.
- The **advanced search** portal to use for more complex queries.

The unified search box interacts with a unified search option implementation as follows:

- As the user types free form text in the search box, the application uses the **search filter rules** and **hints** to parse the entered text and translate it to corresponding query filters.
- The **search service script** is then called to query for entities that match these filters and present the results.
- Selecting a result record navigates to the corresponding portal.
- Selecting the **Advanced Search** link in the search box navigates to the general query portal associated with the unified search option.

If a unified search option is not enabled for your product or the user does not have security access to it then the search box may only be used for searching the menu.

The following sections describe various aspects related to the configuration of a unified search option.

### The Search Script Controls Everything

A unified search is implemented as a single service script that controls the following:

- **Filters.** The script determines which filters are supported and how they are applied to return results efficiently. Each filter name must be a valid **Field** record.
- **Results.** The script returns a list of business entities that matches the query filters provided by the user, along with the navigation information needed to drill in to a selected record in the list.
- **Application security.** Only users that have access to the script's application service may use the unified search associated with the search script.
- **Override Advance Search.** The script may override the advanced search portal configured for the search option.

When the search box component initializes, the script is called once in **Prepare Search Details** mode to return the list of supported filter names. In this mode the script does not perform any query.


Copyright © 2026, Oracle and/or its affiliates.

As the user types in text in the search box, the search box component analyzes the entered text to derive query filters and calls the search script in **Perform Search** mode to return matching results.

The search box component is designed to support any type of unified search implementation and as such, the interaction between the user interface component and the search script is generic. Filters are defined as name / value pairs and results for each matched entity includes details like primary, secondary and tertiary description lines, along with a line that describes the matching text for entry.

> **Note**
>
> If your organization needs to implement a custom unified search script, it is strongly recommended that you follow the base product's implementation of such scripts as a reference.

### Filter Rules and Hints

A unified search is designed to support a predefined set of filters for querying data. For example, searching customers by their address, name, identifiers; or searching locations by address or name etc. Each filter is uniquely identified by a **Field** name as declared by the search service script.

As the user types in free text in the search box, the application applies parsing rules to determine whether the text may be used as the value of one or more filters supported by the unified search. For example, if the user types in the text "ave", parsing rules may determine that it can be used to filter results by a person's name or address but not their email address.

Each parsing rule, defined in the **Search Filter Rule** extendable lookup, associates a filter's field name with a regular expression. Multiple filter rules defined for the same filter name are processed in execution order sequence and it is enough for one rule to return a match for the search text to be used as the value for the specific filter.

Alternatively, the user may use an explicit hint code to identify the filter they would like to use followed by the filter value. For example, an address filter may be associated with the hint "ad:" allowing the user to enter "ad: main" to look up records by address.

Each hint, defined in the **Search Filter Hint** extendable lookup, associates a filter's field name with a hint code. When applicable, a hint may be associated with a value expression rule that is used to qualify the entered value as eligible for use as a filter value. For example, requiring the value to be of a minimum length before a search is performed. If the entered value does not match the hint's value expression the corresponding search is not performed.

Note that filter rules and hints are defined for a specific filter name and not a specific unified search option. The same parsing rules may be shared across searches using the same filter names. For example the parsing rule to identify an "email" filter or the "ad:" hint for an address filter are the same regardless of how these are used by a search query. These rules are automatically associated with any unified search option that uses the same filter names.

> **Note**
>
> You may customize the parsing rules and hints associated with a filter by inactivating the base product's extendable lookup records and introducing custom rules that match your business requirements.


Copyright © 2026, Oracle and/or its affiliates.

### Which Unified Search Option is Used?

Each unified search option is defined in the **Unified Search** extendable lookup, referencing the option's search service script and its corresponding advanced search portal. The unified search option is only enabled to users that have access to the search script's application service. When more than one unified search option is configured and enabled for a user, the application chooses the option with the highest sequence as the default option, allowing users to switch to other options they have access to.

### Customizing Unified Search

You may customize the parsing rules and hints associated with base product search filters by inactivating the base product's extendable lookup records and introducing custom rules that match your business requirements.

There is no need to duplicate a product hint to override its Value Expression rule. You may do so by populating the Override Value Expression element on the hint definition record.

If you need to support a search for different business entities and filters, you need to implement a custom unified search. The configuration of a unified search option involves the following high level steps:

- Design and implement a search script that supports a predefined set of filters and returns the topmost set of matching results while taking into account performance considerations.
- Design hints and filter rules as needed for each of the supported filters. Review existing hints and filter rules associated with same filter names when applicable.
- Register each hint, if any in the **Search Filter Hint** extendable lookup.
- Register each filter rule, if any in the **Search Filter Rule** extendable lookup.
- Register the search script in the **Unified Search** extendable lookup.

# Trees

Many edge products and implementations make extensive use of UI tree structures to display hierarchical relationships between records in the system. The framework provides an objects to define a tree structure and the rules on how to populate the nodes in the tree, and processes to interpret the configuration and dynamically render the tree display.

The topics in this section provide information about what is supplied in the framework. Your product may supply UI components with ready built trees for common use cases. Refer to your product documentation for more information.

### Common Attributes of Trees

Your organization may have many portals and zones showing data in a hierarchical tree. Most trees in the system follow a similar pattern:

- Trees have a number of levels. There may be different types of tree nodes present in any given level; for instance a list of algorithm types for a script, followed by a list of script steps. Icons may be displayed to further identify the node’s object type.
- The most common node types reference an entity within the system such as a schema, script, data area and so on. These nodes usually display the information string from the entity's FK reference and provide the ability to navigate to the entity. They may also allow the node context to be broadcast to other zones in the portal.


Copyright © 2026, Oracle and/or its affiliates.

- In addition to the ability to navigate to the associated entity, many node types provide a context menu with further navigation paths.
- Some nodes are simply informational and are used to group a set of child nodes below a type or category. For example, the business object Options tree groups the option value nodes below summary nodes that describe the option type.

Each tree defines the nodes that make up the tree and the parent/child relationship between those nodes. A tree node defines which of the common attributes are applicable to the node and the algorithms responsible for retrieving the node data and, optionally, determining the node information to be displayed.

### Populate Node Algorithm

A tree node displays a combination of the attributes listed above, typically for an object in context. For a root node, the object will normally be derived from a global or portal context value, such as the current script. For each lower level node, the context is derived from the context fields of its parent row; for instance, algorithms for a specific algorithm type.

The key responsibility of the populate node algorithm is to receive the tree code, node name and the parent context value (or values), determine if there is data that should be shown for this node, and return the a list of the context fields and values applicable to each row in the node. The following highlights additional optional information that may be returned by the algorithm:

- Entity information. If the node is related to a specific entity, the maintenance object code and primary key values may be returned. These are used for determining default information for the node, including navigation details.
- A display icon value. This is used to override the icon defined on the tree node and/or the default icon for the entity as defined on the maintenance object or business object.
- An information string. This may be used to override the default information string for an entity or to display stand-alone text such as a summary or description of the child nodes to follow.
- Override broadcast details if needed for a specific node. For example, set specific broadcast field names and values when they differ from the related entity's primary keys or indicate that broadcasting is not applicable for a node instance even though broadcasting is enabled by the node's definition.

> **Note**
>
> The populate node algorithm is responsible for returning all the context values required to populate its child nodes. For an informational or summary node, this may require including the node’s parent context fields in its own context. For example, in the script tree, the summary node describing the number of steps is followed by a list of steps for the script. The summary node populate node algorithm needs to include the script code passed in as parent context in its output context list, in order for the steps to be correctly retrieved.

### Information Override Algorithm

The information override algorithm has two main purposes:

- Providing alternative information text for a tree node when the information provided isn’t suitable for your implementation.


Copyright © 2026, Oracle and/or its affiliates.

- Retrieving contextual insights related to the tree node entity, to be included in the node's information text. Only badge insight types are currently supported.
- Override broadcast details if needed for a specific node. For example, set specific broadcast field names and values when they differ from the related entity's primary keys or indicate that broadcasting is not applicable for a node instance even though broadcasting is enabled by the node's definition.

The override information algorithm receives the tree code, node name, maintenance object and prime key values for the node. The following highlights the information that may be returned by the algorithm:

- A display icon value.
- An information string. This will be used to override the default information string in the node's formatted string.
- Full node text. This will be used as the full text for the display, overriding the formatting defined on the node.
- A list of badge insight types and their context.

> **Note**
>
> If the information override algorithm returns insight types for the entity, it is also responsible for returning all the context values required for those insight types. Refer to Contextual Insights for more information.

**Preparing a Node for Display**

The following section highlights key steps in fetching a node's data and formatting it for display.

**Determining Node Content**

The following steps are performed to retrieve the components making up the tree node display:

- The Populate Node algorithm is called to determine all instance nodes for the tree node along with their related details.
- The Information Override algorithm is called for each node instance to allow for optional override of presentation related details.

The node display contents are determined as follows:

- The icon is determined from the following sources, in order: tree node definition, override information algorithm value, populate node algorithm value, maintenance object / business object default value.
- Broadcast usage is determined by the setting on the tree node definition.
  - This setting can be overriden by either the Populate Node algorithm or the Override Information algoirthm (which takes precedence) to indicate that broadcasting is not applicable for a specific node instance.
  - By default the broadcast fields for a node are the primary key names of its related entity. Broadcast field names can be overriden by either the Populate Node algorithm or an Override Information algorithm where the latter takes precedence.
- The information string is determined from the following sources, in order: override information algorithm value, populate node value, maintenance object / business object information algorithm. If the string is derived from the MO/BO algorithm, the short information text is used when available else the standard information is used.


Copyright © 2026, Oracle and/or its affiliates.

- Context menu usage is determined by the settings on the tree node definition.
- The system supports a different style for header rows, including a larger font. Header style usage is determined by the setting on the tree definition and is only applicable if the node is a root node with only one instance.

**Node Text Format**

The text for nodes referencing a system entity commonly includes some combination of a label, the entity key and an information string. The node configuration provides a label field and a substitution format string field which may be used to define the node text format. The format string defines which of the components should be included in the text, the order of those components and any surrounding text and HTML markups. For example, a format string of '`%label` - `%key` - `%info`' will result in a node text like the example below:

If the format string is blank, the default text will include '`%info`' only. If an override information algorithm is configured that returns the full node text, the text is presented without further formatting.

**Auto Expansion**

If the tree node definition indicates that the node should be expanded, the system will automatically fetch and display the data for its child node. This processing continues until there are no further child nodes marked for expansion.

There are cases where a tree node should be conditionally expanded based on business rules. For example, prevent the initial expansion when the number of child nodes exceeds a certain threshold. You may design the "Populate Node" algorithm to indicate whether a tree node should be initially expanded or not. When populated, the indication overrides the setting on the tree node definition.

**Base Delivered Trees**

Trees are system data and your product may already deliver several trees out of the box. You may choose to customize a base delivered tree in the following ways:

- You may add your own nodes to an existing tree definition.
- You may override the information string for a node by supplying your own **Information Override** algorithm. If a base override algorithm is configured on the node, your custom algorithm should be plugged in at a higher sequence number, to take precedence over the base algorithm.

**Rendering a Tree**

To render a tree on the UI, you would include the ou-tree web component in a UI map that references the tree definition.

> **Note**
>
> Refer to Include a Web Component for more information.

- Defining Trees


Copyright © 2026, Oracle and/or its affiliates.

## Defining Trees

This portal is used to view and maintain information about a tree definition. Trees are commonly used to display hierarchical relationships between objects in the system. Refer to Trees for background information about defining trees.

You can access the portal from **Admin**, and then **System**, and then **Tree**. You are brought to a query portal with options for searching for a specific tree definition. Once a tree has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones will appear as part of the portal's **Main** tab page

- **Tree.** This zone provides information about the selected tree.
- **Tree Nodes.** This zone lists the nodes for this tree in hierarchical order. Use the zone actions to add or maintain a node in the tree.
- **Tree Node.** This zone displays the details for a tree node. It is only visible if a tree node name has been broadcast from the Tree Nodes zone.

# Contextual Insights

Contextual Insights provide a powerful way to surface focused intelligence and alerts to users in the context of the core screens for any applications built on Oracle Utilities Application Framework. There are multiple options for the visual patterns when surfacing Contextual Insights and the actions that can be performed by users are easily configurable as well. The subsequent topics provide detail about the Insight Type and Insight Group objects that are used to configure the information that may then be displayed in appropriate screens.

- Insight Types
- Insight Groups

## Insight Types

The insight type object contains all the detailed configuration for displaying one or more alerts, or important information for a record in context. The following sections provide more information about all the configuration managed by the insight type.

### Insight Algorithm

Each insight type is used to display some information, typically related to an object in context. It is the responsibility of the insight type algorithm to receive the context value (or values), determine if there is information that should be shown for this context, return the text to display.

The following highlights additional optional information that may be returned by the algorithm:

- A secondary message. This is only applicable to some visual structures where more than one line of information may be displayed.
- A severity value. This is used to override the visual display configuration if different emphasis should be used for different severity values. Refer to Override Configuration for more information.
- Entity information. If the insight is related to a specific entity, the maintenance object code and primary key values may be returned. This is useful if the insight type configures a BPA or user action algorithms that may perform logic for this entity.


Copyright © 2026, Oracle and/or its affiliates.

- Navigation and User Action information. This is only applicable to visual structures that support user actions. The algorithm must return the list of all the actions that should be made visible on the insight. If the action includes navigation, the algorithm should return the navigation option and navigation context key and value pairs. Additionally, there is configuration for each action that must be defined. Refer to User Actions for more information.

Each algorithm can return multiple insights, if applicable. For example, if the insight is about To Do entries related to the object being displayed, you could design your insight algorithm to return a single insight summarizing all the To Do related information. Or you could choose to return separate insights based on To Do information so that you provide more granular information and customize actions or navigation based on the different conditions. For example, if there are To Do Entries that the current user may perform actions on and others that current user cannot update, the algorithm could return two separate insights: one with actions and one without.

> **Note**
>
> **Preview mode**. The insight type maintenance page includes an Insight Preview section that shows the output from the insight algorithm when invoked with an insight mode of "Preview". Every algorithm should include logic that returns appropriate text when called in this mode so that something is displayed in that section.

### Visual Structures

Each insight type defines a visual structure which dictates what kind of information may be shown. The system has some fixed layout expectations for each visual structure type. The insight type configuration is used to provide the source for the possible layout.

> **Note**
>
> All the insights use color options as part of their configuration. Refer to Color Option for more information.

### Badge Insight

The simplest visual structure is the badge. It is used to display a single line of quick text with a solid background (for visibility) that is meant to alert the user to important summary information or attributes.




### Inline Insight

The inline insight defines an icon and a single message. This is meant to be more descriptive than a badge. For this type of insight, a default light background is used so the recommendation is to select a color option for both the text and the icon that has only a foreground color defined or has a light background color.

### List Insight

The list insight are intended to exist in a list of insights for the object. It defines an icon and can support two lines of text. In addition, this type of insight supports buttons or user actions. Refer to User Actions for more information. Note that for a list insight, if there is one valid action or 2


Copyright © 2026, Oracle and/or its affiliates.

valid actions, the action buttons are shown. If there are three or more valid actions, one action is shown and then an 'Other Actions' button is visible. Clicking this shows the other available actions.

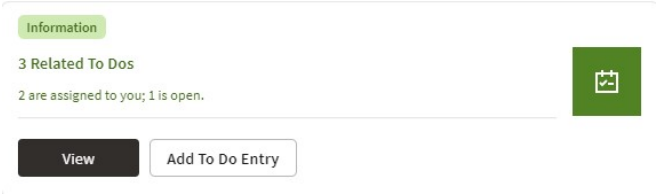
Note that the background color for the insight is used as the color for the full insight snippet. The text and the icon are shown on this background. The icon's background color is ignored.

### Card

The card insight is the most information rich, with up to two lines of text, user actions, a large icon to call attention and header text. The header text is chosen from a dropdown list of valid options. The product provides out of the box header text of "Information", "Warning" and "Error". Implementations may add additional values to this using the lookup field HEADER_CAT_FLG.

Please note the following about the treatment of the various components of the card insight:

- The header text is rendered like a badge insight.
- For the message text displayed in the card, the background color is ignored so a color option with no background color or a light background color should be chosen.
- For the icon, the background color is used to create a square area for display and the icon is rendered in the foreground color.




### Override Configuration

Each insight supports default visual setting for the various components of the insight. This includes the color option for the applicable components along with icons and header text, where applicable. In each case, override visual settings may be defined for a given severity number.

For example imagine the algorithm for a badge insight that returns a message with the count of related To Do entries uses a different color option if there are more than 10 related records. To do this, the algorithm can return a severity number and then the configuration user can define the override color option for this severity number.

### User Actions

The list and card insights support the definition of user actions that may be presented as buttons when displaying the insight. The actions may do one or more of the following:

- Perform algorithm logic. One or more user action algorithms may be associated with an action button. This should be used to perform server logic that does not require any user interaction. For example, you can provide an action to update the status of a record related to the insight. These algorithms are configured on the Insight Type.


Copyright © 2026, Oracle and/or its affiliates.

- Navigate to another page. For this type of action, the insight algorithm must provide the navigation context and the navigation option to launch when clicking the button. It's possible to configure navigation and configure user action algorithms. In that case the algorithms are run first and then the navigation occurs.
- Launch a BPA. This is used if there is user interaction needed to perform additional logic and / or if there is conditional navigation such that the logic is needed before navigating. It's possible to configure a BPA and configure user action algorithms. In that case the algorithms are run first and then the BPA is launched. If the insight algorithm also provides navigation for this action, it is ignored in favor of the BPA. The BPA is configured on the Insight Type.

The list of valid actions to display at run time must be returned by the insight algorithm. This allows for the algorithm to have control over which actions to display, in case there are some actions that are conditional. In addition, each possible button requires configuration related to what text should display and the style of the button.

**Button Text**

The button text to display is chosen from a button text lookup. The lookup name is `INSIGHT_ACT_FLD_FLG`. Implementations may add additional values to this lookup to include new button text. In addition, if you would like to change button text on any product delivered button, you can override the lookup value description.

**Action Class**

Each action button defines an action class. Any action configured with the action class value of **Positive** value is shown first and is styled differently for emphasis. Other action class values are for information purposes.

**Securing Actions**

If the action navigates or launches a BPA, the system automatically checks the underlying security for the target page or the BPA and only shows the action if the user has security rights. Other types of actions, such as ones that should launch an algorithm, do not have security configuration that can be automatically detected by the system. For these types of actions, you can define an explicit application service and access mode on the Buttons collection. It's also possible for the algorithm to check security before returning applicable buttons.

**Base Delivered Insight Types**

Insight types are system data and your product may already deliver several insight types out of the box. You may choose to include one or more insight types in an Insight Group.

**Rendering an Insight**

To render an insight for an insight type, you would include the ou-insights web component in a UI map that references this insight type. This is how the Insight Preview section of the insight type maintenance page renders the preview.

> **Note**
>
> Refer to Include a Web Component for more information.

- Defining Insight Types


Copyright © 2026, Oracle and/or its affiliates.

## Defining Insight Types

The insight type entity is used to define the configuration of a specific type of alert to display extra information related to one or more records in context. Refer to Insight Types for a detailed description of the supported functionality.

To view or maintain insight types, open **Admin**, and then **General**, and then **Insight Type.**

This is a standard All-in-One portal.

The information captured on the insight type depends on the record's business object. Refer to the embedded help text for more information.

**Related Insight Groups**

If there are any insight groups that reference this insight type, the **Related Insight Groups** zone displays the linked insight groups.

# Insight Groups

An insight group may be used to include several insight types together onto a single map. When defining an insight group, you configure the visual structures that the insight group should include. Only insight types for the configured visual structure are available.

**Insight Class**

Each insight group must reference a unique insight class. The insight class is defined using a lookup. The lookup field is `INSIGHT_CLASS_FLG`. Your product may already deliver base values for insight class. To display insights in a user interface your UI map would include the ou-insights web component. This component includes syntax to reference an insight class and the logic will find the insight group for that class and render all the insight types linked to that group.

Because the insight class is used as the reference in the web component rather than the insight group, it allows your product to supply base logic to show insight group information in a base delivered user interface. The base logic defines an insight class in base and references it on a base UI map. Your implementation can then define the insight group for this insight class and choose the insight types that should be rendered in that map. Refer to your specific product documentation to find out if there are any base delivered insight classes.

If your implementation identifies a new use case where one ore more insight types should be displayed, once the insight types are configured, do the following:

- Define a new Insight Class lookup value.
- Define an Insight Group that references this insight class and configure the appropriate insight types.
- Include the appropriate web component in the map for your user interface where you wish to display this information.

**Multiple Insight Groups for an Insight Class**

Typically there should exist a single insight group for an insight class. In cases where the list of insight types for the class needs to vary based on context, a separate insight group needs to be defined for each list.


Copyright © 2026, Oracle and/or its affiliates.

When more than one insight group is associated with an insight class **Action Method** rules may be applied to determine which group to use based on the context provided to insight processing. Refer to Action Method for an overview of action method functionality.

- Defining Insight Groups

## Defining Insight Groups

The insight group entity allows implementations to define multiple related insight types that should be displayed together in a given user interface map or zone. Refer to Insight Groups for an overview of insight group configuration and functionality.

To view or maintain insight groups, open **Admin**, and then **General**, and then **Insight Group.**

This is a standard All-in-One portal.

The information captured on the insight group depends on the record's business object. Refer to the embedded help text for more information.

**Related Insight Types**

The **Related Insight Types** zone displays all the insight types that are currently linked to the group. The Edit link in the zone header allows you to add or remove insight types. Only insight types that reference one of the **Valid Visual Structures** may be selected.

# Knowledge Articles

When applicable, a knowledge article icon may be configured to appear in a specific location on a page. Clicking the icon launches a drawer presenting an article on the topic and allows the user to further search and review other articles.

> **Note**
>
> The product supports only knowledge articles defined in Oracle Fusion.

The association between a user interface location and a related knowledge article is registered in the **Knowledge Management Articles** extendable lookup. The user interface location references an extendable lookup knowledge code and the article ID is referenced on the extendable lookup record. The knowledge article icon appears on the page only when the corresponding extendable lookup entry is active and references an article ID.

Some product pages are released with embedded references to knowledge codes. By default, these knowledge codes are not associated with any article. To enable these locations for knowledge functionality update their respective extendable lookup entries with references to your articles.

You may also introduce knowledge functionality to custom pages as follows:

- Add a knowledge code entry to the extendable lookup along with a reference to an article ID.
- Reference the knowledge code on your custom page using the Knowledge Article web component.

Access to knowledge articles requires an integration with Oracle Fusion. Once the integration is established, follow these instructions to complete the configuration:

- Update the context details for the **Get Fusion Article** Message Sender.


Copyright © 2026, Oracle and/or its affiliates.

**Table Get Fusion Article - Message Sender (F1-KMGET)**

| Context Type | Context Value |
|---|---|
| HTTP Method | GET |
| HTTP Timeout | 60 |
| HTTP Transfer Method | SendReceive |
| HTTP URL 1 | @EXT_PUB@<hostname>:<portnumber>/api/knowledge-management/v2/articles/answers/${pathParms}?${queryParms} |
| OAuth Grant Type | jwt_user_assertion |
| OAuth Access Token URL<br>OAuth Client Credentials Key Ring<br>Oauth Signature Key Ring<br>OAuth REST API Scope | Refer to OAuth Configuration for more information. |

- Update the context details for the **Search Fusion Articles** Message Sender.

**Table Search Fusion Articles - Message Sender (F1-KMSCH)**

| Context Type | Context Value |
|---|---|
| HTTP Method | GET |
| HTTP Timeout | 60 |
| HTTP Transfer Method | SendReceive |
| HTTP URL 1 | @EXT_PUB@<hostname>:<portnumber>/api/knowledge-management/v2/search?${queryParms} |
| OAuth Grant Type | jwt_user_assertion |
| OAuth Access Token URL<br>OAuth Client Credentials Key Ring<br>Oauth Signature Key Ring<br>OAuth REST API Scope | Refer to OAuth Configuration for more information. |

# Color Configuration

The topics in this section describe some options for configuring colors to be used in displaying information in the user interface.

- Defining Colors
- Color Options

## Defining Colors

The system provides the extendable lookup Color (F1-Color) to define valid colors to be used in the system. The base colors provided are ones related to the standard color palette used in Oracle products.


Copyright © 2026, Oracle and/or its affiliates.

At this time, zones and UI maps that define specific colors do not support any syntax to reference a color extendable lookup. In the meantime, colors applied at the zone or map level can still reference HTML colors or reference hex codes directly.

## Color Options

The color option entity is used to define a combination of colors to be used in user interface components. The system provides a standard set of color options used for contextual insights.

To view or maintain color options, open **Admin**, and then **System**, and then **Color Option.**

This is a standard All-in-One portal.

The information captured on the color option depends on the record's business object. Refer to the embedded help text for more information.

# Defining Display Icons

Icons are used to assist users in identifying different types of objects or instructions. A limited number of control tables allow administrative users to select an icon when they are configuring the system. Select **Admin**, and then **System**, and then **Display Icon** to maintain the population of icons available for selection. You are brought to a query portal with options for searching for a specific display icon. Once a display icon has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The latest standard is to use Scalable Vector Graphic (SVG) icons. For these types of icons, the Icon Type is **Variable (SVG)**. The product also supports icons in the .gif or .png format. For these types of icons, the Icon Type defines a pixel size.

**URL** describes where the icon is located (folder) and the icon name (file name).

Note that there is a standard folder structure for delivered assets such as images. The structure is `{DefaultWebApp}/{owner}/assets/images|svgs/`

- {Owner} - this is the owner of the icon. The `/cm` folder is used for custom icons.
- svgs - all images that are of type svg are in this folder.
- images - all files that have a file suffix of jpg, gif, png or other types of image formats are in this folder.

When using the above folder structure, then the URL only needs to contain the icon name (file name). The correct full URL is determined automatically at runtime. Images can optionally be placed in other folders under the DefaultWebApp. However, this is not recommended but if used, the URL must contain the full path to the icon (e.g. /cm/assets/newImages/myIcon.gif)

# Custom Look and Feel Options

The default look and feel of the application can be customized via feature configuration. The base product is provided with a **Custom Look And Feel** Feature Configuration type. If you define a feature configuration of this type, you can adjust the look and feel in the following ways.

- You may adjust the look and feel of the application using cascading style sheets. Refer to User Interface Style Customization for more information.
- You may adjust configuration related to the display of the UI Map Help icon.


Copyright © 2026, Oracle and/or its affiliates.

- You may choose to change the UI View Settings for all users in the system. To do this, define an entry for the **UI View** option type and enter the desired value. Note that individual users may override this setting on their user preferences.
- Redwood Enablement
- User Interface Style Customization
- UI Map Help

## Redwood Enablement

The Redwood User Experience is opt-in functionality. To enable this in your implementation, navigate to Feature Configuration. Search for an existing feature configuration with the Feature Type `Custom Look and Feel`. If one does not exist, create a feature configuration of this type. Select the `User Interface Style` option and set the value to 'Oracle Redwood'.

## User Interface Style Customization

The base product allows for the conditional inclusion of custom style sheets into the system style set. Custom styles may override any style provided by the base product. The style sheet may also include new styles for use in customer zone definitions. Use the **Style Sheet** option on the **Custom Look And Feel** Feature Configuration to define your custom style sheet. The option supports either referencing a *.css file or referencing a Managed Content record (with the content type **CSS Definitions**.)

> **Note**
>
> The Managed Content option is recommended as this makes the Custom CSS available to be easily transferred to other environments if desired using Content Migration Assistant (CMA).

> **Note**
>
> Some styles cannot change if they are part of the HTML code.

> **Caution**
>
> Implementers must ensure that the customized user interface is stable and scalable. Changing font, alignment padding, border size, and other user interface parameters may cause presentation problems, like scrollbars appearing or disappearing, cursors not working as expected, and unanticipated look and feel alterations of some layouts.

The structure of the CSS extension follows the standard CSS file format regardless of using the File option or the Managed Content option. This means that the "<style>" tag is not needed and the file will contain a list of CSS selectors with their declarations (properties and values).

To find the necessary CSS Selector, navigate the application to an appropriate screen, then "right click" on the element you wish to re-style and select **inspect** from the menu (normally at the bottom). This opens a view of the HTML document. From this view you will be able to see any "id" and "class" attributes that are available on the element. These are generally the most suitable attributes for CSS selectors. In order to make the styling more specific, it is also


Copyright © 2026, Oracle and/or its affiliates.

appropriate to scan up the HTML DOM tree to view parent "id" and "class" attributes. Making the CSS selector as precisely targeted as possible will reduce the possibility of custom stylings affecting elements outside your desired area.

All standard CSS is supported. However, when looking at the most up-to-date CSS standards, be aware that not all browsers support all options without some specific browser settings.

## UI Map Help

A tool tip can be used to display additional help information to the user. This applies to section elements as well as individual elements on a map zone or UI Map. Refer to the tips context sensitive zone associated with the UI Map page for more information. The **Custom Look And Feel** Feature Configuration provides options to control the following:

- Whether UI Map Help functionality is turned on or off. By default it is turned on.
- Override the default help image with a custom image
- The location of the help image, either before or after the element.

> **NOT_SUPPORTED**
>
> Refer to the feature configuration for a detailed description of each option.

# Database Options

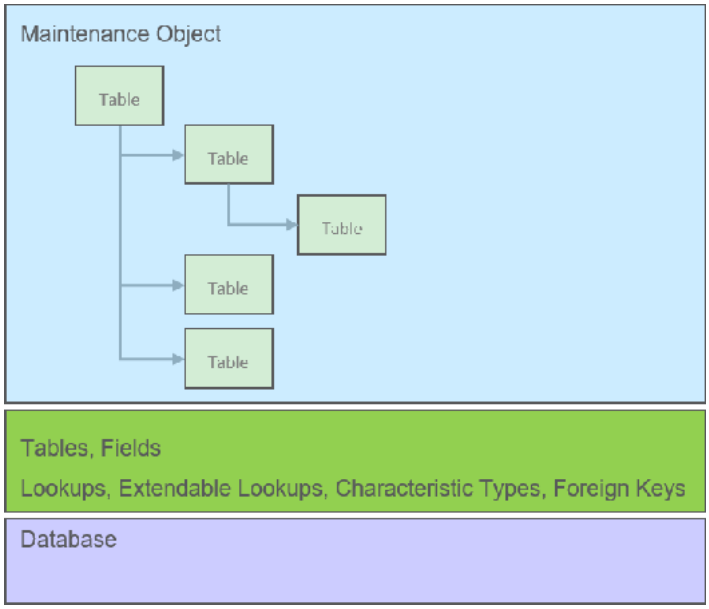
This section describes a variety of database tools that are supplied with the your product.

- Understanding The Data Model
- Defining Tables
- Defining Fields
- Defining Maintenance Objects
- Maintenance Object Configuration
- Defining Lookups
- Defining Extendable Lookups
- Defining Characteristic Types
- Defining Foreign Key References
- Audit Trails
- Bundling
- Revision Control
- Information Lifecycle Management

## Understanding The Data Model

The application uses meta-data to describe its data model, from plain description of its tables and fields to more complex hierarchies of maintenance objects.


Copyright © 2026, Oracle and/or its affiliates.




The following points briefly outline key aspects of application's data model as illustrated in this diagram:

- Table and Field records describe all the tables maintained by the application.
- Valid values for certain fields are defined using corresponding Lookup and Extendable Lookup records.
- Characteristic types may be defined to support an extension of a maintenance object via its characteristics table.
- Maintenance Objects define the physical structure of an entity. Business objects further refine maintenance objects to describe business entities. Refer to The Big Picture of Business Objects for more information.
- Foreign Key References define ways for searching and navigating to the maintenance portal of an entity.

The following sections further discusses concepts and guidelines related to the data model.

- Fields
- Tables
- Maintenance Objects
- Defining Valid Values
- Characteristic Types
- Foreign Key References

# Fields

A Field definition describes the metadata needed by the application to support the capture and display of a piece of information. Fields may correspond to physical columns on database tables or represent any type of element used for calculation or presentation purposes.

The definition includes the following key details:

- Whether the field refers to a physical column on a database table or used as a work field.
- The type of data the field holds.


Copyright © 2026, Oracle and/or its affiliates.

- The label to use when the field is presented.
- The help text to display when the user clicks on the help icon seeking additional information about this field.
- and more

The following sections further discusses concepts and guidelines related to defining fields.

### Physical vs. Work Fields

A Field definition record exists for every physical column on a database table that is used by the application. Refer to Understanding Tables for more information.

A Field that does not represent a physical column on a database table is considered as work field and needs to be marked as such. Work fields are typically used as labels as well as describe internal elements used in business rules.

### Data Type Information

The type of data captured by a field is defined using the following attributes:

- **Data Type** indicates the basic type of value the field holds. Valid values are **Character, Character Large Object, Date, DateTime, Number, Time, Varchar2** and **XML Type**.
- **Extended Data Type** is used to further define the type of data for certain data types. Valid values are **Currency Source, Day of Month, Duration, Money, Month of Year, Flag, Switch** and **URI**.
- **Precision** defines the length of the field. In the case of variable length fields, it is the maximum length possible. For number fields that include decimal values, the precision includes the decimal values.
- **Scale** is only applicable for number fields. It indicates the number of decimal places supported by the field.
- **Sign** is only applicable for numbers. It indicates if the data may contain positive or negative numbers.

### Data Type Inheritance

A Field can be defined to inherit its data type attributes from the definition of another field. The field to inherit the attributes from is referenced as the **Base Field**. When using a base field reference, the data type attributes are defined once, on the base field, and all the fields inheriting their data type information from it are kept consistent with that definition as it may change over time. Any change of a field’s data type attributes is automatically reflected on all the fields that reference it as their base field.

Data type information may either be specified directly on the field itself or by referencing a base field. The system does not support a way to partially override the base field’s data type information.

It is recommended to leverage data type inheritance whenever possible, especially when the reason for creating a new field is to provide an alternate label for the original field in a certain context. In that situation the new field should reference the original field as its base field and provide the new labels and help text as needed.

### Labels and Help Text

For fields that appear on a user interface, the description acts as a label to be used for presentation purposes. As per the product's multi-language support, a field is associated with a label for each of the languages enabled in the system.


Copyright © 2026, Oracle and/or its affiliates.

A field's help text may be used to provide the user with additional information on how the field is used. When populated for a field that is visible on a user interface that supports display of embedded help, a help icon appears for the field, presenting the help text to the user when clicked.

### Overriding Labels and Help Text

The label and help text provided for a product owned field may be overridden by your organization using the corresponding override fields.

The field's label and help text can be overridden for a specific table by populating the corresponding override information as part of the table / field definition. However, this override is not used in portal based user interfaces. It is only applicable if the field is displayed on fixed page user interfaces. For fixed pages, if the override label at the table / field level takes precedence over the override at the field level.

### Alternate Labels

There are contexts where a separate label may be needed for the "date" and "time" parts of a date/time field. Such an example is when the field is used by the operational analytics product. For example, for a "Completion Date/Time" field you may provide the context label "Creation Date" for the **Date** context and "Creation Time" for the **Time** context. These context labels are optional.

In the same way, date fields, time fields and date/time fields may be associated with a **Date and Time Base** context label, which is a label without a reference to the word "Date", "Time" or "Date / Time". For example, for a "Completion Date/Time" field, the date and time base label may be "Completion". This label that can be used in combination with date or time attributes such as day, month, year, hours, minutes etc.

Overrides labels are available for these additional context labels.

## Tables

A Table definition describes the metadata needed by the application to work with and access a database table or view.

The definition includes the following key details:

- The table's description in the application.
- The list of fields defined on the table and any foreign key constraints they establish with other tables.
- The table's prime key and how it is generated.
- Whether the table holds administrative, master or transactional data.
- Whether the table is released with product owned system data rows or not.
- The table's role in the conversion process.
- How should table content be upgraded.
- and more

The following sections further discusses concepts and guidelines related to defining tables.

### System Tables

Tables released with rows that are owned by the base product are referred to as system tables. These tables include the **Owner** (OWNER_FLG) field to capture the owner of each row.


Copyright © 2026, Oracle and/or its affiliates.

Refer to System Data for more information about naming conventions and standards around row record ownership.

The **Upgrade** setting on the table definition record controls what happens to the rows in a table when the system is upgraded to a new release. The most common setting allows customer owned rows to be merged with rows owned by the base product. Other settings may keep the content of the table as is, i.e. rows are not touched by the upgrade, or fully refresh the content of the table with base product rows.

Some fields on certain system owned tables are explicitly marked to allow customization. This means implementation are allowed to change these specific fields for a base owned record. Changes to the field value of one of these types of fields by an implementation are maintained when upgrading to a new version of the product.

### Designated Table Roles

Some tables are designed to support the following application roles:

- **Key Tables**
  - A key table holds a copy of a table's prime key only and is only applicable to tables with system generated keys.
  - Tables with a system generated keys reference their Key Table as part of their Table definition record.
  - Records in the key table remain in the system even when the related entity is archived, preventing the same key from being allocated to a new record. Retaining the key allows the system to keep the integrity of any references that may still exist to the archived entity.
- **Language Tables**
  - To support multi-language implementations, all labels and descriptions that are presented to a user are kept in a designated language table where a row exists for each enabled language.
  - The product is released with the English content provided in the various language tables. An additional row is added to these tables for each enabled language along with the translated content for that language. Refer to Support For Different Languages for more information.
  - Tables associated with language sensitive fields reference their corresponding Language Table as part of their Table definition record.
- **Characteristics Tables**
  - Many entities in the system support one or more Characteristics tables, which are used to capture additional fields for the object that are not already supported by the entity's provided attributes.
  - A Characteristics table is designed to capture characteristics applicable to its specific entity only. The entity associated with a Characteristics table is referenced on the Table's definition record.
  - Refer to Setting Up Characteristic Types & Values for more information.
- **Entity Log Tables**
  - Many entities in the system support the ability to capture lifecycle and other important business events on a designated log table.
  - A log table captures the event details in the form of an application message and stores any parameters for the message in a child log parameters table. All log and log


Copyright © 2026, Oracle and/or its affiliates.

parameters tables follow a predefined structure allowing them to be consistently and efficiently implemented.

  - Typically log tables do not participate in the conversion process.

### Table Constraints

Foreign key relationships between tables are not enforced at the database level but rather at the application level, providing the application more flexibility around data manipulation and referential integrity.

The following types of constraints may be defined on the Table's definition record:

- **Primary Key** represents the field or set of fields that represent the unique identifier of a record stored in a table.
- **Logical Key** represents an alternate unique identifier of a record based on a different set of fields than the Primary key.
- **Foreign Key** represents a field or set of fields that specifies identifying and non-identifying relationships to other tables in the application. A foreign key constraint references the primary key constraint of another table.
- **Conditional Foreign Key** represents rare relationships between tables where a single field (or set of fields) may reference multiple primary key constraints of other tables within the application as a foreign key.

The system leverages a table's foreign key constraints information to performs referential integrity validation when rows in this table are deleted. These validations can be disabled by checking off the **Enable Referential Integrity** field on the Table definition record.

### Table Indexes

The indexes delivered by the base product for a given table are visible in the index metadata. In addition, table index information may be defined as part of the cloud services support for generation of custom tables in certain use cases. Refer to the section Generating Custom Tables below for more information.

### Primary Key Type

A table's primary key may be of the following types:

- **User-defined** means the user specifies the key when a record is added. This type of key is typically used for configuration and administrative tables.
- **System-generated** means a program generates a random numeric key for the record when it is added.
  - This type of table is typically associated with a key table. Refer to the **Designated Table Roles** section for more information.
  - System generated keys may be fully random or partially inherited from a parent table's key for performance reasons. Refer to the **System Generated Keys** for more information.
- **Sequential** means a sequence number is incremented whenever a record is added to the table. The next number in the sequence determines the key value.
- **Database Native Sequential Key** means a unique number is assigned using a database sequence object.
- **Other** means some other business rule allocates the table's prime-key or the prime key is made of multiple parts.


Copyright © 2026, Oracle and/or its affiliates.

### System Generated Keys

In general, randomly generated numeric system keys are used to attempt to evenly distribute records across a full range of possible IDs. Batch programs that use multiple threads will typically divide the threads using ID ranges and evenly distributed keys will help spread out the work across these threads.

If a table using a system generated key is a child of another table, the child table's key may **inherit** a portion of the random number from its parent table's key. Key inheritance for child records further extends the performance benefit. When considering partitioning, the recommendation for DBAs is to range partition data based on the primary key so that different batch threads operate on different partitions which reduces contention for hot blocks. Ideally the number of batch threads will be an exact multiple of the number of partitions. Batch programs that insert child data (for example batch Billing creation) also benefit from this design especially when the child tables are partitioned in the same way. The parent is often the driver of the batch process. If this is multi-threaded, then each thread is processing a set of parent records in a given ID range and all child records are being inserted into the same ID range.

While most entities with system generated keys either use fully random keys or partially inherited keys from a single parent, entities may be designed to implement more specific key generation rules.

When key inheritance is used, the table definition of the child table defines the following information:

- The number of most significant digits from the parent record's primary key to use as the **prefix** for the child record's key value.
- The **parent key field name** to inherit the key from. This information is provided for display and documentation purposes only and is only provided when the inheritance rule implemented by the entity may be documented this way. Typically the child key value is inherited from a single parent key but in some cases the inheritance rule may be more complex. For example, different types of records in the same child table may inherit their key from different parent key fields. In this case, multiple parent field names may be listed to describe the various inheritance options. When the rule is too complex, this information may not be provided on the table definition record.

### Conversion Options

The conversion process manages the process by which master and transaction legacy data, loaded into a staging schema, is validated and assigned new system generated keys before it is copied to the production schema. Refer to Understanding The Conversion Process for more information.

The tool classifies tables into the following roles:

- Administrative tables. These only exist in the production schema. Instead of a table, a view to the production table exists in the staging schema. These views are necessary to enforce foreign key references when data is validated in the staging schema
- Convertible tables that retain their prime keys. Data conversion is supported for these tables and their prime key values remain unchanged.
- Convertible tables that require a new system generated key to be assigned. Data conversion is supported for these tables and their prime key is reassigned by the system during conversion.
- Not convertible tables. Assigning new system generated keys for a record and replacing all foreign key references it might have with the new key is a complicated process that may not be supported for all tables.


Copyright © 2026, Oracle and/or its affiliates.

. The **Data Conversion Role** setting on the table definition record controls if / how the table is used by conversion processing.

### Date/Time Configuration

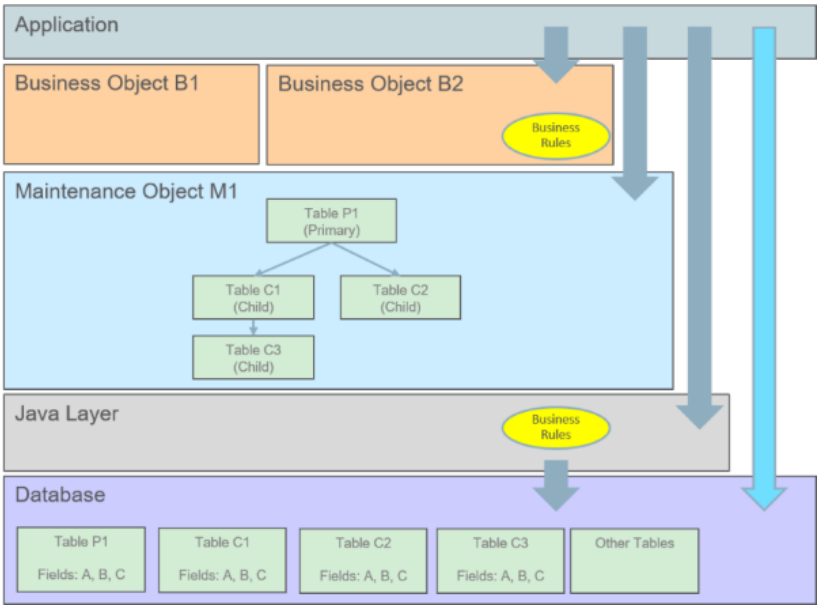
Fields capturing date/time information may be designed to store data in either local legal time or local standard time. Local legal time is captured as entered and displayed as captured in the database. It also assumes the date/time information refers to the time zone defined on the installation options. Refer to Standard vs. Legal for more information.

### Generating Custom Tables

For cloud customers and for certain use cases, the product supports defining table metadata for custom tables and generating the related DDLs for that metadata in the database. For these custom tables, there is a Generation Status visible on the Table metadata that identifies where the Table is in the lifecycle for this supported feature. In addition, the index tab on the table portal is used to define custom indexes that are included in the DDL as part of this feature. If you are a cloud customer and would like more information, refer to the Cloud Services Implementation Guide.

## Maintenance Objects

A maintenance object defines the configuration of a given “entity” in the system. It includes a primary table and any additional child tables that together capture the physical data for the entity. The application reads, adds, updates and deletes an entity as a single object by invoking its maintenance object service.




The following points briefly outline key aspects of application’s interaction with maintenance objects as illustrated in this diagram:

- Database tables are maintained solely via the application’s Java layer. Direct updates to a table via its Java class is not common and typically reserved to special processes.
- A maintenance object orchestrates the update to an entity’s primary table and its child tables as a single object.
- A business object defines a logical representation of an entity, including all or a subset of the maintenance object’s tables and fields.


Copyright © 2026, Oracle and/or its affiliates.

- The more common way to maintain an entity is via its maintenance object, either directly or via one of its business objects. Most entities support the use of business objects and as such their maintenance is typically made via their business objects.
- Direct access to database tables is supported for query purposes only.

The following sections further discusses concepts and guidelines related to defining tables.

#### Maintaining An Entire Object

A maintenance object includes a single primary table and one or more child tables where each child table can further have nested child tables. The table structure of a maintenance object represents the scope of data that should be maintained together as a single object:

- When an object is added, data in added to each corresponding table.
- When an object is updated, data in updated in each corresponding table.
- When an object is read, data in retrieved from all the tables.
- When an object is deleted, data in deleted from all the tables.

#### Algorithms

Several algorithm plug-in spots are also defined on the maintenance object, allowing for business rules that govern all records for this maintenance object.

#### Options

The maintenance object includes options that define important information related to the maintenance object that may be accessed for logic throughout the system.

#### Maintenance Objects vs. Business Objects

A maintenance object defines the physical structure of an entity as stored in the database. Different business objects may be defined on top of a maintenance object to represent different types of such entity. A business object includes only the relevant information applicable to the specific type of entity as well as enforces business rules applicable only to that type.

When a maintenance object supports business object configuration, the system uses the **Determine Business Object** maintenance object algorithm to determine the identifying business object that controls a given entity's data structure and rules.

Refer to The Big Picture of Business Objects for more information.

## Defining Valid Values

The product provides several options for defining valid values for a column on a table:

- Lookup
- Extendable Lookup
- Control Table

The following provides more information about the functionality of each of the options available for defining valid values for a column.

#### Lookup

The simplest mechanism for defining valid values for a column on a table is via the Lookup table. This is sometimes referred to as a "simple" lookup to distinguish it from an extendable lookup (described below). Using the lookup table, you can define valid values and their


Copyright © 2026, Oracle and/or its affiliates.

descriptions. When choosing a valid value that is defined by a lookup, a dropdown UI metaphor is used.

The following highlights functionality related to lookups:

- Lookups are associated with a Field. The field is defined as a character data type with an extended data type of **Flag**. The field's label serves as the description for the prompt to select the valid value.
- The lookup code is limited to four characters and must be all uppercase. If there is any functionality where a valid value in the application must match valid values in an external system, the lookup table may not be the appropriate choice.
- The lookup table does not support additional attributes to be defined for each value. This option is only appropriate when a simple code and description pair is needed.
- The product may also use Lookups to define valid values for functionality unrelated to a column on a table. For example, an algorithm plug-in spot may define an input parameter that supports one or more valid values. The plug-in spot may define the valid values using a lookup, allowing for a simple way to validate the value supplied when invoking the algorithm and to document the valid values.

> **NOT_SUPPORTED**
>
> For more information, refer to Defining Lookup Options.

#### Extendable Lookup

The extendable lookup provides a way of defining valid values for a column with additional capabilities that are not supported using the Lookup table. When choosing a valid value that is defined by an extendable lookup, a dropdown UI metaphor is used.

The following highlights functionality related to extendable lookups:

- Each Extendable Lookups is defined using a business object.
- A field should be defined for the extendable lookup code. The field defines the label for the lookup code and defines the size of the lookup code. The size is determined based on the business use case. In addition, there are standard fields included in all extendable lookups, including a description, detailed description and an override description (so that implementations can override the description of base delivered values).
- The extendable lookup may define additional information for each value if warranted by the business requirement. See Additional Attributes for technical information about additional attributes.

> **NOT_SUPPORTED**
>
> For more information, refer to Defining Extendable Lookups.

#### Control Table

There may be scenarios where a list of valid values warrants a standalone maintenance object, which is considered an administrative or control table object. When choosing a valid value that is defined by a control, either a dropdown UI metaphor or a search metaphor is used, depending on how it has been designed.

The following points highlight some reasons why this option may be chosen:


Copyright © 2026, Oracle and/or its affiliates.

- The records require a lifecycle such that BO status is warranted.
- The additional attributes are sophisticated enough that they warrant their own column definition rather than relying on using CLOB or flattened characteristic. For example, if a list of information needs to be captured with several attributes in the list and the information in the list needs to be searchable.

In this situation, if a product has provided a control table for this type of functionality, it will be documented fully in the appropriate functional area. If an implementation determines that a custom control table is warranted, all the standard functionality for a maintenance object is required: database tables, maintenance object metadata, appropriate Java maintenance classes, portals, zones, etc. Refer to the Software Development Kit for more information. No further information is provided in this section for this option.

# Characteristic Types

Many maintenance objects in the system support characteristics tables, which are used to capture additional fields for the object that are not already supported by the object's provided attributes. Each characteristic is associated with a characteristic type, which defines attributes of the field you wish to capture.

While each characteristic is stored in a separate row in the characteristic table, the user interface metaphor used to capture and display them differ based on the type of page that is used to maintain the object.

- For portal based pages, the business object drives the display and maintenance of the object. The display and maintenance of each characteristic is determined by the business object's user interface design. For these types of pages, it is recommended that characteristics are defined as flattened elements allowing the user interface to display the characteristic as if it is another field. Refer to The Flattening Nodes and Attributes for more information about flattening characteristics on a business object schema.
- There are some fixed pages in the system that do not support customization of the user interface. For these objects, the characteristics are displayed / maintained as a generic list.

The topics in this section describe various aspects of a characteristic type definition and usage.

### Characteristic Value Types

Every characteristic referenced on an object references a characteristic type. The characteristic type controls the validity of the information entered by a user when they enter the characteristic's values. For example, if you have a characteristic type on user called "skill", the information you setup on this characteristic type controls the valid values that may be specified by a user when defining another user's skill.

When you setup a characteristic type, you must classify it as one of the following categories:

- **Predefined Value.** When you setup a characteristic of this type, you define the individual valid values that may be entered by a user. A good example of such a characteristic type would be one on User to define one or more predefined skills for that user. The valid values for this characteristic type would be defined in a discreet list.
- **Adhoc Value.** Characteristics of this type do not have their valid values defined in a discreet list because the possible values are infinite. Good examples of such a characteristic type would be ones used to define a user's birth date or their mother's maiden name. Optionally, you can plug-in an algorithm on such a characteristic type to validate the value entered by the user. For example, you can plug-in an algorithm on a characteristic type to ensure the value entered is a date.


Copyright © 2026, Oracle and/or its affiliates.

- **Foreign Key Value.** Characteristics of this type have their valid values defined in another table. For example perhaps you want to link a user to a table where User is not already a foreign key. Valid values for this type of characteristic would be defined on the user table. Before you can create a characteristic of this type, information about the table that contains the valid values must be defined on the foreign key reference table.

  > **Note**
  >
  > When you are searching for a foreign key characteristic value, there are some limitations of when a search is available depending on the search metaphore. Refer to Search Options for more information.

- **File Location.** Characteristics of this type contain a URL. The URL can point to a file or any web site. Characteristics of this type might be useful to hold references to documentation / images associated with a given entity. For example, the image of a letter sent to you by one of your customers could be referenced as a file location characteristic on a customer contact entry. When such a characteristic is defined on an entity, a button can be used to open the URL in a separate browser window.

  File location characteristic values must be entered in a "non-relative" format. For example, if you want to define a characteristic value of *www.msn.com*, enter the characteristic value as `http://www.msn.com`. If you omit the `http://` prefix, the system will suffix the characteristic value to the current URL in your browser and attempt to navigate to this location when the launch button is pressed. This may or may not be the desired result.

  > **Note**
  >
  > Due to browser security restrictions, opening URLs using the file protocol ("file://") from pages retrieved using http does not work. If the file protocol is used, the browser either does not return properly or an error is thrown (e.g., "Access Denied", which usually results from cross site scripting features added for security reasons). This issue has no known workaround. To comply with browser security standards, the recommendation is to move the target files to an FTP or HTTP server location to avoid protocols that are subject to browser security restrictions.
  >
  > Also note that the functionality described in the topics for Referencing URIs do not apply to this value given that the browser is responsible for connecting to the URI and does not go via server logic.

  For references to a file, the recommendation is to use the Attachment functionality to link a file to an object rather than a characteristic type of File Location. Refer to Attachment Overview for more information. The documentation related to file location remains for upgrade purposes.

### Searching By Characteristic Values

For certain entities in the system that have characteristics, you may search for a record linked to a given characteristic value. The search may be done in one of the following ways:

- Some base searches provide an option to search for an object by entering Characteristic Type and Characteristic Value.
- Your implementation may define a customized search for an entity by a characteristic value for a specific characteristic type using a query data explorer.


Copyright © 2026, Oracle and/or its affiliates.

- Your implementation may require a business service to find a record via a given characteristic value. For example, maybe an upload of user information attempts to find the user via an Employee ID, defined as a characteristic.

Not all entities that support characteristics support searching by characteristics. To confirm if a given characteristic table supports searching, verify that it includes the field SRCH_CHAR_VAL.

> ⚠ **Caution**
>
> For ad hoc characteristics, only the first 50 bytes are searchable. For foreign key characteristics, the search value is populated by concatenating the values of each foreign key column to a maximum of 50 bytes.

For the base searches that provide a generic option to search by characteristic type and value, you can restrict the characteristic types that can be used to search for an entity. For example, imagine you use a characteristic to define a "jurisdiction" associated with a To Do for reporting purposes. If your company operates within a very small number of jurisdictions, you wouldn't want to allow searching for a To Do by jurisdiction, as a large number of To Do entries would be returned.

A flag on the characteristic type allows an administrator to indicate if searching by this characteristic type is **allowed** or **not allowed**.

#### Data Typing an Ad hoc Characteristic Value

An **Adhoc Value** characteristic type may further restrict the data type of the captured value by referencing a Field. When populated, the system validates that the captured characteristic value complies with the data type definition of the referenced field.

#### Valid Characteristic Entities

Characteristics are stored in designated tables, each associated with a unique characteristic entity value referenced on the table definition record. A characteristic type record lists all the valid characteristic entities it is allowed to be captured on.

#### Customizing Base Owned Characteristic Types

Implementations may customize the following aspects of a base owned characteristic type:

- Extend the list of valid values for a base owned **Predefined Value** characteristic type to include additional custom values. Such extension is only supported if the characteristic type is defined with the **Custom** option turned on.
- Provide a custom algorithm to validate an **Adhoc Value** characteristic type.
- Indicate whether characteristics of this type may be used in searches.
- Extend the list of entities associated with the characteristic type.

## Foreign Key References

A Foreign Key Reference defines the necessary information needed to reference an entity in a certain table.

You need to set up this control table if you need to validate a foreign key value against a corresponding table. For example, if a schema element is associated with an FK Reference the system validates the element's value against the corresponding table. Refer to Configuration Tools to learn more about schema-based objects. Another example is characteristics whose


Copyright © 2026, Oracle and/or its affiliates.

valid values are defined in another table (i.e., you use "foreign key reference" characteristic types). Refer to Characteristic Types for a description of characteristics of this type.

A FK Reference is used not just for validation purposes. It also used to display the standard information description of the reference entity as well as provide navigation information to its maintenance transaction. Info descriptions appear throughout the UI, for example, whenever an account is displayed on a page, a description of the account appears. The product provides base product FK references for many of its entities as they are used for validation and display of elements in both fixed page user interfaces as well as portal based user interfaces.

An implementation may also see the need to define a foreign key reference. The following points describe what you should know before you can setup a foreign key reference for a table.

- The physical name of the table. Typically this is the primary table of a maintenance object.
- The program used by default to construct the referenced entity's info description.
- The transaction used to maintain the referenced entity. This is where the user navigates to when using the "go to" button or hyperlink associated with the entity.
- The name of the search page or zone used to look for a valid entity.

The topics in this section describe various aspects of a foreign key reference definition and usage.

### Information Description Is Dynamically Derived

Typically a FK Reference is defined for a maintenance object's primary table. In this case the system dynamically derives the standard information associated with a specific referenced entity as follows:

- Attempt to determine the business object associated with the referenced entity. Refer to the Determine BO maintenance object algorithm system event for more information. If a business object has been determined, the system lets the business object's Information plug-in, if any, format the description.
- If a business object has not been determined or the business object has no such plug-in, the system lets the maintenance object's information plug-in, if any, format the description.
- If the maintenance object has no such plug-in, the system uses the info program specified on the FK Reference to format the information.

> **Note**
>
> **Technical note.** The class that returns the information displayed adjacent to the referenced entity is generated specifically for use as an info routine. Please speak to your support group if you need to generate such a class.

> **Note**
>
> **Generic routine.** The system provides a generic information routine that returns the description of control table objects from its associated language table. By "control table" we mean a table with an associated language table that contains a **DESCR** field. Refer to Defining Table Options for more information on tables and fields. The java class is `com.splwg.base.domain.common.foreignKeyReference.DescriptionRetriever`.


Copyright © 2026, Oracle and/or its affiliates.

### Navigation Information Is Dynamically Derived

Typically a FK Reference is defined for a maintenance object's primary table. In this case the system dynamically derives the actual transaction to navigate to for a given referenced entity as follows:

- Attempt to determine the business object associated with the referenced entity. Refer to the Determine BO maintenance object algorithm system event for more information. If a business object has been determined, use the maintenance portal defined as its **Portal Navigation Option** business object option.
- If a business object has not been determined or the business object defines no such option, the system uses the transaction specified on the FK Reference.

### Search Options

The product provides two main metaphors for implementing a user interface. For input fields that are foreign keys, search options are dependent on the metaphor used by the page in question.

- A portal based user interface is a more flexible user interface where an implementation has more options for customizing the look and feel. The base product uses UI maps or automatic UI rendering to display input fields. Elements that are foreign keys may display a search icon if the FK reference defines a Search Zone.

  > **Note**
  >
  > **Defining search zones directly.** It's possible for elements on a UI map to define a specific search zone directly in the HTML, rather than using the search zone defined on an FK reference. Refer to the UI map tips for more information on implementing searches using zones.

- A fixed maintenance page user interface is a page supplied by the base product where only minor enhancements, if any, can be introduced by implementations. The foreign key reference may be used in one of two ways.
  - The based product may use an FK reference to define a base element on one of these pages. If a search is available for such elements, the FK reference's Search Navigation Key is used to implement the search.
  - Entities that support characteristics typically include a generic characteristic collection UI metaphor on these types of pages. In this metaphor, a foreign key characteristic displays a search icon if the FK Reference has configured a Search Navigation Key.

> **Note**
>
> Not every FK reference provided with the product is configured with a search option. This may be the case if the base delivered pages use a dropdown for this foreign key rather than a search. In addition, base provided FK references that do provide a search may not be configured with both search options. It means that if linking this type of object as a characteristic, the search may not be available if the appropriate search is not configured.


Copyright © 2026, Oracle and/or its affiliates.

# Defining Tables

This portal is used to view and maintain information about an application table.

You can access the portal from the**Admin Menu**, and then **Database**, and then **Table**. You are brought to a query portal with options for searching for a specific table. Once a table has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Tables for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Table.** This zone provides information about the selected table.
- **Fields.** This zone lists all fields of the selected table and provides various actions to maintain this information.

The following zones may appear as part of the portal's **Relationship** tab page

- **Entity Relationship Diagram.** This zone provides a visual representation of the table's relationship to other tables in the application's data model.

The following zones may appear as part of the portal's **Constraints** tab page

- **Table Constraints.** This zone lists the various key constraints defined for the selected table and provides various actions to maintain this information.
- **Referred By Constraints.** This zone lists all tables referring to the selected table via foreign key constraints.

Refer to Tables for more information.

The following zones may appear as part of the portal's **Indexes** tab page

- **Table Indexes.** This zone lists the indexes defined for the selected table and provides various actions to maintain this information.

Refer to Tables for more information.

# Defining Fields

This portal is used to view and maintain information about a field. For product owned fields, you may use this portal to override the label of a field on the various pages in the system.

You can access the portal from the **Admin Menu**, and then **Database**, and then **Field**. You are brought to a query portal with options for searching for a specific field. Once a field has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Fields for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Field.** This zone provides information about the selected field.


Copyright © 2026, Oracle and/or its affiliates.

- **Tables Using Field.** This zone lists tables referencing the field. It is not visible for work fields.
- **Lookup Values.** This zone lists and maintains valid values for the selected lookup field. The zone is only visible when the selected field is a lookup field.

The following zone may appear as part of the portal's **References** tab page

- **Field References.** This zone list various configuration entities referencing the current field.

# Defining Maintenance Objects

A maintenance object defines the configuration of a given “entity” in the system. It includes the definition of the tables that together capture the physical data for the entity. In addition, the maintenance object includes options that define important information related to the maintenance object that may be accessed for logic throughout the system. Several algorithm plug-in spots are also defined on the maintenance object, allowing for business rules that govern all records for this maintenance object.

Most maintenance objects are provided with the base package. An implementation can introduce custom maintenance objects when needed. Most fields may not be changed if owned by the base package.

Many maintenance objects in the system support the use of business objects to further define configuration and business rules for a given record. Refer to Maintenance Objects for more information.

You can access the portal from the **Admin Menu**, and then **Database**, and then **Maintenance Object**. You are brought to a query portal with options for searching for a specific maintenance object. Once a maintenance object has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Maintenance Object.** This zone provides information about the selected maintenance object.
- **Tables Information.** This zone displays information about the tables included in the maintenance object.

The following zone may appear as part of the portal's **Relationship** tab page

- **Entity Relationship Diagram.** This zone describes the relatiomship of this entity to other entities in the data model.

The following zone may appear as part of the portal's **Options** tab page

- **Options.** This zone is used to maintain various configuration options associated with the current maintenance object.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **You can add new option types.** Your implementation may want to add additional maintenance option types. For example, your implementation may have plug-in driven logic that would benefit from a new type of option. To do that, add your new values to the customizable lookup field `MAINT_OBJ_OPT_FLG`.

The following zone may appear as part of the portal's **Algorithms** tab page

- **Algorithms.** This zone is used to maintain the algorithms associated with the current maintenance object. Refer to the table that follows for a description of all possible system events. If the algorithm is implemented as a script, a link to the **Script** is provided. Refer to Plug-in Scripts for more information.

The following zone may appear as part of the portal's **Business Objects** tab page

- **Business Objects.** This zone displays business objects associated with the current maintenance object.

The following zone may appear as part of the portal's **References** tab page

- **Maintenance Object References.** This zone list various configuration entities referencing the current maintenance object.

### Actions

The page supports the following special actions:

- **Create an Algorithm.** Use this action, on the Algorithm tab to create a script based algorithm related to this maintenance object. You are prompted for information regarding the plug-in spot and the system event, the name, description, etc. Once all the information is provided, the system creates an algorithm type, a script, an algorithm and links the algorithm to the maintenance object. You may then navigate to the script to adjust it as needed.
- **Inactivate / Activate Base Algorithms.** You can inactivate algorithms on Maintenance Objects. Your implementation may want to use a maintenance object provided by the base product, but may want to inactivate one or more algorithms provided by the base maintenance object. To do that, use the "Inactivate" action on the appropriate algorithm entry on the Algorithm tab. Use use the "Activate" action to mark an inactive algorithm as active.
  Note that these actions add and remove the "Inactive Algorithm" maintenance option as needed.
- Maintenance Object Algorithms

# Maintenance Object Algorithms

The following table describes all the system events supported for maintenance objects.


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| **Audit** | Optional | Algorithms of this type are called to notify of any changes to the maintenance object's set of tables. These algorithms are invoked just before the commit at the end of a logical transaction. The system keeps track of what records are added or changed in the course of a transaction and all MO audit algorithms are executed in order of when each record was first added or updated. |
| **Determine BO** | Optional | Algorithm of this type is used to determine the Business Object associated with an instance of the maintenance object. It is necessary to plug in such an algorithm on a Maintenance Object to enable the business object rules functionality.<br>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number. |
| **ILM Eligibility** | Optional | Algorithms of this type are used for maintenance objects that are enabled for object erasure for Information Lifecycle Management. They are used to review records that have reached the maximum retention days and evaluate if they are ready to be archived.<br>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
| --- | --- | --- |
| **Information** | Optional | We use the term "Maintenance Object Information" to describe the basic information that appears throughout the system to describe an instance of the maintenance object. The data that appears in this information description is constructed using this algorithm.<br>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number. |
| **Manage Erasure Schedule** | Optional | Algorithms of this type are used for maintenance objects that are enabled for object erasure, which is a measure to protect Data Privacy. They are triggered when certain system events occur for those objects. The algorithms are used to determine if the record needs to be scheduled for erasure and, if so, create or maintain an entry for the record in the Object Erasure Schedule. |
| **Revision Control** | Optional | An algorithm of this type is used to enforce revision control rules when an object is added, changed, or deleted. The maintenance object service calls the plug-in once before the object is processed and once more after applying all business object rules. This allows revision rules to take place in proper revision timings. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| **Transition** | Optional | The system calls algorithms of this type upon each successful state transition of a business object as well as when it is first created. These are typically used to record the transition on the maintenance object's log.<br>Note that most base maintenance objects are already shipped with an automatic logging of state transitions. In this case you may use these algorithms to override the base logging functionality with your own. Refer to State Transitions are Audited for more information. |
| **Transition Error** | Optional | The system calls this type of algorithm when a state transition fails and the business object should be saved in its latest successful state. The algorithm is responsible for logging the transition error somewhere, typically on the maintenance object's log.<br>Notice that in this case, the caller does NOT get an error back but rather the call ends successfully and the exception is recorded somewhere, as per the plug-in logic.<br>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number. |

# Maintenance Object Configuration

The Maintenance Object Configuration portal provides several zones to help implementations manage configuration across several maintenance objects. It is a standard All-in-One portal.

Navigate using**Admin**, and then **Implementation Tools**, and then **Maintenance Object Configuration.**

The following zones appear on the portal.

- **Object Erasure Configuation.** This zone displays a list of maintenance objects that support object erasure. Press the edit icon to open a window in which the configuration values for that maintenance object can be changed.


Copyright © 2026, Oracle and/or its affiliates.

- **Generalized Initial Export Batch Controls.** This zone displays a list of maintenance objects that are eligible for generalize data export, but where appropriate batch controls have not been created. Note that base delivered maintenance objects are delivered with the appropriate batch controls. This zone may not show any data unless you have custom maintenance objects that should be configured for Generalized Data Export.
- **Conversion Batch Controls.** This zone displays a list of maintenance objects whose primary table is eligible for conversion, but where appropriate batch controls have not been created. Note that base delivered maintenance objects are delivered with the appropriate batch controls. This zone may not show any data unless you have custom maintenance objects that should be configured for conversion.

# Defining Lookups

Lookup fields may be used to define valid values for a column in a table or for other types of values like parameters to an algorithm. This portal is used to view and maintain information about a lookup field and its values. Note that all lookup information is also maintained as part of the corresponding Field portal.

You can access the portal from the **Admin Menu**, and then **Database**, and then **Lookup**. You are brought to a query portal with options for searching for a specific lookup field. Once a field has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **NOT_SUPPORTED**
>
> Refer to Defining Valid Values for some background information.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Lookup.** This zone provides information about the selected lookup field.
- **Lookup Values.** This zone lists and maintains valid values for the selected lookup field.

The base product provides many different lookup fields and their values as part of the product. The following points highlight some functionality related to base-package lookups.

- Fields that are owned by the product will typically provide base lookup values. Implementations are not permitted to remove base delivered lookup values. Implementations may be able to add custom values to base owned lookups. This is controlled with the Custom switch on lookup.
  - When the custom switch is unchecked, it means that there is functionality controlled by the base values and an implementation may not extend or customize this functionality. An example of this type of lookup is the Data Type field on the Field table. The system supports a distinct list of data types and an implementation may not add additional values.
  - When the custom switch is checked, it means that there is base functionality supplied for the base values but that an implementation can extend the functionality by supplying their own values. An example of this type of lookup is the Access Mode on


Copyright © 2026, Oracle and/or its affiliates.

Application Service. The product provides many values for the access mode lookup, representing various actions a user may perform. Implementations may add their own values to this lookup. Documentation should indicate when functionality may be extended and should highlight the lookup value that can be extended.

> ⚠ **Caution**
>
> Important! If you introduce new lookup values, you must prefix the lookup value code with X or Y. If you do not do this, there is a possibility that a future release of the application could introduce a new lookup value with the name you allocated.

- There may be some scenarios where the product supplies a base field and base lookup field with no base lookup values supplied. This occurs when the product doesn't have any base functionality driven by the lookup values. Typically this type of lookup is for information or categorization purposes. The configuration guide for the functional area associated with the lookup should include a configuration step regarding defining values for this type of lookup.
- The description of base delivered values may be overridden by an implementation.

An implementation may also identify the need for defining a new lookup field with its values.

# Defining Extendable Lookups

Extendable lookups are a way of defining valid values that are more sophisticated than simple lookups.

> **NOT_SUPPORTED**
>
> Refer to Defining Valid Values for some background information.

The base product provides extendable lookups as part of the product. The following points highlight some functionality related to base-package extendable lookups.

- The base product may supply base extendable lookup values. Implementations are not permitted to remove base delivered extendable lookup values. It is also possible that implementations may be able to add custom values to base owned lookups. If an implementation is not permitted to add lookup values to the base extendable lookup, the extendable lookup business object will include validation to prevent this. There is no equivalent of the Custom switch that is on the lookup field.
- There may be some scenarios where the product supplies a base extendable lookup with no base lookup values supplied. This occurs when the product doesn't have any base functionality driven by the extendable lookup values. The configuration guide for the functional area associated with the extendable lookup should include a configuration step regarding defining values for this type of extendable lookup.
- The description of base delivered values may be overridden by an implementation.

Open this page using **Admin**, and then **General**, and then **Extendable Lookup.**

You are brought to the **Extendable Lookup Query** where you need to search for the extendable lookup object (in other words, its business object).


Copyright © 2026, Oracle and/or its affiliates.

Once you have found the appropriate extendable lookup, select the value and you are brought to a standard All-in-One portal that lists the existing lookup values for the extendable lookup. The standard actions for an All-in-One portal are available here.

- Extendable Lookup Advanced Topics

# Extendable Lookup Advanced Topics

This section provides some addition technical information about extendable lookup attributes

### Defining Additional Attributes

The product provides a few different ways to define additional values for an extendable lookup. Some of the methods are only relevant for base delivered lookup values as they may impact whether or not an implementation can update the values.

The following table highlights the options available and some summary information about what the option provides.

| Option | Brief Description | Extendable Lookup Value Searchable by this Attribute? | Base Delivered Value Modifiable? |
|---|---|---|---|
| Element mapped to BO_DATA_AREA | The element is mapped to a CLOB field that allows for base delivered values to be modified. | No | Yes |
| Element mapped to BASE_BO_DATA_AREA | The element is mapped to a CLOB field that does not allow for base delivered values to be modified. | No | No |
| Flattened characteristic | The element is defined using the flattened characteristic mechanism. | Yes | No |

The following points highlight information from the table above:

- The decision of defining an additional attribute using a CLOB mapping or a flattened characteristic will depend on whether the functionality expects that the lookup value is known when the attribute is needed (in which case a CLOB mapping is appropriate) or if the functionality expects to determine the lookup value based on the attribute (in which case, a flattened characteristic is appropriate).
- When the base product defines an extendable lookup with additional attributes and intends to provide base extendable lookup values, it needs to determine whether or not implementations may update the additional attribute or not.
  - If no and the value is mapped to a CLOB, it will map the value to the BASE_BO_DATA_AREA column. This means that implementations will receive an owner mismatch error when attempting to change the value. In addition, upgrading to a new release will replace the value with the base value.
  - If yes and the value is mapped to a CLOB, it will map the value to the BO_DATA_AREA column. This means that implementations will be able to change the value for a base owned record. In addition, upgrading to a new release will not make any changes to the value.


Copyright © 2026, Oracle and/or its affiliates.

- For values mapped to a characteristic, the product does not support an implementation changing the value of a base delivered record. If the product would like to support an implementation overriding this type of value, the business object will need to be designed with a corresponding "override" element (also a flattened characteristic), similar to how the product supplies an Override Description field to support an implementation overriding the base product delivered description for a base value. This element will not be delivered with any value and will allow an implementation to populate that value.

  > **Note**
  >
  > Note that in this situation, the product functionality that uses this value must cater for the override value.

- All of this detail is only relevant for base provided extendable lookup values. If an implementation adds custom values for a base supplied extendable lookup, all the additional attributes may be populated as appropriate.
- If an implementation defines a custom extendable lookup business object and wants to define an additional attribute using a CLOB, it doesn't matter which CLOB column is used. Both BO_DATA_AREA and BASE_BO_DATA_AREA provide the same functionality for custom business objects.

**Capturing a Password**

If an extendable lookup includes configuration of a password for some functionality, the system supports automatic encryption of the password value if the schema maps the password to a characteristic using the characteristic type `F1-PWD`.

# Defining Characteristic Types

This portal is used to view and maintain characteristic types. An implementation may define their own characteristic types or override some of the settings for base product provided characteristic types.

You can access the portal from the **Admin Menu**, and then **General**, and then **Characteristic Type**. You are brought to a query portal with options for searching for a specific record. Once a characteristic type record has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> When introducing a new record, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Characteristic Types for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Characteristic Type.** This zone provides information about the selected characteristic type.
- **Characteristic Values.** This zone is visible for pre-defined characteristic types and shows the existing values (if applicable) with action buttons. It also allows you to add new values.


Copyright © 2026, Oracle and/or its affiliates.

Note that you may only add values to a product owned characteristic type if the Custom switch is checked.

# Defining Foreign Key References

The topics in this section describe the transaction that allows you to define foreign key references.

Refer to Foreign Key References for more information.

You can access the portal from the **Admin Menu**, and then **Database**, and then **FK Reference**. You are brought to a query portal with options for searching for a specific foreign key reference. Once a foreign key reference has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a new foreign key reference, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Foreign Key Reference.** This zone provides information about the selected foreign key reference.
  - Navigation to the referenced entity may be dynamic. The page associated with the specified Navigation Option is used when a dynamic option is not available. Refer to Navigation Information Is Dynamically Derived for more information.
  - The information description of the referenced entity may be composed dynamically depending on the specified Info Program Name. Refer to Information Description Is Dynamically Derived for more information.
  - Certain restrictions apply when displaying the context menu for a referenced entity.

    > ⓘ **Note**
    >
    > Context Menu Name is not applicable to user interface elements displaying a generic collection using a foreign key characteristic type. It is only applicable for pages utilizing the foreign key compound element type for fixed page user interface and for data displayed in a portal based user interface where the foreign key reference is defined as an attribute for an element. Report parameters that reference foreign key characteristics are an example of a user interface where a context menu is not displayed even if the foreign key reference defines one.

  - Certain restrictions apply to search features on fixed page user interfaces.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Search Type and Search Tooltip attributes are only applicable to user interface elements utilizing the foreign key compound element type on fixed page user interfaces. Report parameters that reference foreign key characteristics are an example of a user interface where this information is not used even if the foreign key reference defines them.

The following zones may appear as part of the portal's **References** tab page.

- **Foreign Key References.** This zone list various configuration entities referencing the current foreign key reference.

#### Actions

The page supports the following special actions:

- **View the source.** If the foreign key information program is shipped with the base package, you can use the adjacent button to display the source code of this program in the Javadocs viewer.

# Audit Trails

The topics in this section describe one way of auditing changes in the system. Note that this technique has limitations and may not be the best option for all situations.

- This functionality is configured at the table and field level. Many base maintenance objects use an XML or CLOB field to capture one or more elements using XML format, configured using a BO schema. This auditing technique is not able to capture changes to individual elements. It can only capture overall changes to the single field. The BO Audit plug-in is a better option for auditing changes to individual elements with the CLOB / XML field.
- The base table provided for supporting audits limits the field size of the before and after values to 254 bytes.
- Auditing is captured for each field. In some cases it is prefered to capture a before and after image for several fields at once. For example, if an address is changed, it's more user friendly to capture the before and after for the full address rather than the individual address components. The BO Audit plug-in allows for capturing a single audit record for multiple elements rather than granular changes for each element.

The subsequent topics highlight how to enable auditing for fields, and describe the auditing queries that you can use to view audit records.


- Captured Information
- How Auditing Works
- The Audit Trail File
- How To Enable Auditing
- Audit Queries


## Captured Information

When auditing is enabled for a field, the following information is recorded when the field is changed, added and/or deleted (depending on the actions that you are auditing for that field):

- User ID



Copyright © 2026, Oracle and/or its affiliates.

- Date and time
- Table name
- Row's prime key value
- Field name
- Before image (blank when a row is added)
- After image (blank when a row is deleted)
- Row action (add, change, delete)

## How Auditing Works

You enable auditing on a table in the table's meta-data by specifying the name of the table in which to insert the audit information (the audit table) and the name of the program responsible for inserting the data (the audit trail insert program). Then you define the fields you want to audit by turning on each field's audit switch in the table's field meta-data. You can audit fields for delete, insert and update actions.

Once auditing is enabled for fields in a table, the respective row maintenance program for the table assembles the list of changed fields and calls the audit trail insert program. If any of the changed fields are marked for audit, the audit program inserts audit rows into the audit table.

> **Note**
>
> **Customizing Audit Information.** You may want to maintain audit information other than what is described in Captured Information or you may want to maintain it in a different format. For example, you may want to maintain audit information for an entire row instead of a field. If so, your implementation team can use the base audit program and base audit tables as examples when creating your own audit trail insert program and audit table structures.

## The Audit Trail File

Audit log records are inserted in the audit tables you define. The base product contains a single such table (called `CI_AUDIT` ). However, the base audit insert program is designed to allow you to use multiple audit tables.

If you want to segregate audit information into multiple tables, you must create these tables. Use the following guidelines when creating new audit tables (that use the base delivered audit insert program):

- The new audit tables must look identical to the base table (`CI_AUDIT`).
- The new tables must be prefixed with **CM** (e.g., `CM_AUDIT_A`, `CM_AUDIT_B`, etc.). An appropriate java table name must be defined and an *_impl.java program for that java table name must be defined.
- The name of the new table must be referenced on the various tables whose changes should be logged in the new table.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> It's important to note if you use your own tables (as opposed to using the base package table called `CI_AUDIT`), the SQL used to insert and access audit trail records in the base delivered audit program is dynamic. Otherwise, if the base package's table is used, the SQL is static.

# How To Enable Auditing

Enabling audits is a two-step process:

- First, you must turn on auditing for a table by specifying an audit table and an audit trail insert program.
- Second, you must specify the fields and actions to be audited for the table.

The following topics describe this process.

- Turn On Auditing For a Table
- Specify The Fields and Actions To Be Audited

## Turn On Auditing For a Table

In order to tell the system which fields to audit, you must know the name of the table on which the field is located. You must specify the audit table and the audit trail insert program for a table in the table's meta-data.

> **Note**
>
> Most of the system's table names are fairly intuitive. For example, the user table is called SC_USER, the navigation option table is called CI_NAV_OPT, etc. If you cannot find the table using the search facility on the Table portal, please contact customer support.

To enable auditing for a table:

- Navigate to the Table portal and find the table associated with the field(s) for which you want to capture audit information.
- Specify the name of the **Audit Table**.

> **Note**
>
> **Specifying the Audit Table.** You can use the audit table that comes supplied with the base package ( `CI_AUDIT`) to audit multiple tables and fields. All the audit logs are combined in a single table ( `CI_AUDIT`). However, you can also have a separate audit table for each audited table. Refer to The Audit Trail File for more information.

- Specify the name of the **Audit Program** . The product provides two programs. `com.splwg.base.domain.common.audit.DefaultTableAuditor` is a standard table audit program. In addition


Copyright © 2026, Oracle and/or its affiliates.

`com.splwg.base.domain.common.audit.ModifiedTableAuditor`is supplied as an alternative. This program does not audit inserts or deletes of empty string data.

> **⚠ Caution**
>
> By default, none of a table's fields are marked for audit. Even though you have enabled auditing for a table, you must still specify the fields and actions on those fields to be audited (see below).

## Specify The Fields and Actions To Be Audited

The system only audits actions (insert, update and delete) made to fields that you want audited.

To specify the fields and actions to be audited:

- Navigate to the Table portal for a table on which you have enabled auditing.
- For each field you want to audit, specify the actions you want to audit by turning on the **Audit Delete**, **Audit Insert** and **Audit Update** switches as appropriate.

> **ⓘ Note**
>
> You can also turn on the audit switches using the Field zone at the bottom of the Table portal.

> **⚠ Caution**
>
> Audit Program Caching! The audit program from the table meta-data is read into a program cache on the application server whenever the date changes or when the server starts. If you implement new auditing on a table, your audit trail does not become effective until this program cache is reloaded. In other words, new audits on tables where the audit program was not previously specified do not become effective until the next day (or the next restart of the application server). However, if you change the fields to be audited for a table where the audit program is already in the cache, your changes are effective immediately.

# Audit Queries

There are two queries that can be used to access the audit information.

- Audit Query by User
- Audit Query by Table / Field / Key

## Audit Query by User

This transaction is used to view changes made by a user that are stored on a given Audit Trail File.


Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> The system only audits changes that you've told it to audit. Refer to The Big Picture Of Audit Trails for more information.

Navigate to this page by selecting **Admin**, and then **Database**, and then **Audit Query By User**.

**Description of Page**

To use this transaction:

- Enter the **User ID** of the user whose changes you wish to view.
- Enter the name of the table on which the audit trail information is stored in **Audit Table**. Refer to The Audit Trail File for more information about this field.

> **ⓘ Note**
>
> **Default Note**. If only one audit table is used to store audit trail information, that table is defaulted.

- Specify a date and time range in **Created between** to restrict the records that result from the query.

> **ⓘ Note**
>
> **Default Note**. The current date is defaulted.

- Click the search button to display all changes recorded on a specific audit table associated with a given user.

Information on this query is initially displayed in reverse chronological order.

The following information is displayed in the grid:

- **Row Creation Date** is the date and time that the change was made.
- **Audited Table Name** contains the name of the table whose contents were changed.
- **Primary Key** is the prime key of the row in the **Audited Table** whose contents where changed.
- **Audited Field Name** is the name of the field that was changed.
- **Audit Action** indicates whether the row action was **Add**, **Change** or **Delete**.
- **Field Value Before Update** contains the content of the field before the change. This column is blank if information was **Added**.
- **Field Value After Update** contains the content of the field after the change. This column is blank if information was **Deleted**.

## Audit Query by Table / Field / Key

This transaction is used to view audited changes made to a given table.


Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> The system only audits changes that you've told it to audit. Refer to The Big Picture Of Audit Trails for more information.

This transaction can be used in several different ways:

- You can view all audited changes to a table. To do this, enter the **Audited Table Name** and leave the other input fields blank.
- You can view all audited changes to a given row in a table (e.g., all changes made to a given user). To do this, enter the **Audited Table Name** and row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**).
- You can view all audited changes to a given field in a table (e.g., all changes made to all customers' rates). To do this, enter the **Audited Table Name** and the **Audited Field Name**.
- You can view all audited changes to a given field on a specific row. To do this, enter the **Audited Table Name**, the **Audited Field Name**, and row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**).

Navigate to this page by selecting **Admin**, and then **Database**, and then **Audit Query By Table/Field/Key**.

**Description of Page**

To use this transaction:

- Enter the name of the table whose changes you wish to view in **Audited Table Name**.
- If you wish to restrict the audit trail to changes made to a specific field, enter the **Audited Field Name**.
- If you wish to restrict the audit trail to changes made to a given row, enter the row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**). These fields are dynamic based on the **Audited Table Name**.
- Specify a date and time range in **Created between** to restrict the records that result from the query.

> **ⓘ Note**
>
> The current date is defaulted.

- Click the search button to display all changes made to this data.

Information on this query is initially displayed in reverse chronological order by field.

The following information is displayed in the grid:

- **Create Date/Time** is the date / time that the change was made.
- **User Name** is the name of the person who changed the information.
- **Primary Key** is the prime key of the row in the **Audited Table** whose contents where changed.
- **Audited Field Name** is the name of the field that was changed.
- **Audit Action** indicates whether the row action was **Add**, **Change** or **Delete**.


Copyright © 2026, Oracle and/or its affiliates.

- **Value Before Update** contains the content of the field before the change. This column is blank if information was **Added**.
- **Value After Update** contains the content of the field after the change. This column is blank if information was **Deleted**.

# Bundling

The topics in this section describe the bundling features in the application.


- About Bundling
- Configuring Maintenance Objects for Bundling
- Working with Bundles


## About Bundling

Bundling is the process of grouping entities for export or import from one environment to another.

For example, you might export a set of business objects and service scripts from a development environment and import them into a QA environment for testing. The group of entities is referred to as a bundle. You create export bundles in the source environment; you create import bundles in the target environment.

Working with bundles involves the following tasks:

- Configuring entities for bundling if they are not preconfigured
- Creating an export bundle, which contains a list of entities to be exported from the source environment
- Creating an import bundle to import those entities to the target environment
- Applying the import bundle, which adds or updates the bundled entities to the target environment


- Sequencing of Objects in a Bundle
- Recursive Key References
- Owner Flags on Bundled Entities


### Sequencing of Objects in a Bundle

Bundle entities are added or updated to the target environment in the sequence defined in the bundle

Typically, the sequence of entities does not matter. However, sequence is important in the following situations:

- Entities that are referenced as foreign keys should be at the top of the sequence, before the entities that reference them. Specify zones last, as they typically contain numerous foreign key references.
- When importing a business object, specify the business object first, then its plug-in scripts, then the algorithms that reference the scripts, and then the algorithm types that reference the algorithms.
- When importing a portal and its zones, specify the portal first and then its zones.



Copyright © 2026, Oracle and/or its affiliates.

- When importing a multi-query zone, specify the referenced zones first and then the multi-query zone.
- Always specify algorithms types before algorithms.

You can specify the sequence when you define the export bundle or when you import the bundle to the target environment.

## Recursive Key References

Recursive foreign keys result when one object has a foreign key reference to another object that in turn has a foreign key reference to the first object.

For example, a zone has foreign keys to its portals, which have foreign keys to their zones. If the objects you want to bundle have recursive relationships, you must create a 'bundling add' business object that has only the minimal number of elements needed to add the entity. A bundling add business object for a zone contains only the zone code and description, with no references to its portals. In the same way, a bundling add business object for a portal defines only its code and description.

When you apply the bundle, the system initially adds the maintenance object based on the elements defined in the bundling add business object. Before committing the bundle, the system updates the maintenance object with the complete set of elements based on its physical business object.

Note that use of the bundling add BO also benefits records that have optional foreign keys or foreign keys that are part of a child table. That way the person creating the bundle does not have to worry about the sequence of the records. For example, an FK Reference may optionally reference a zone for searching. If a new FK Reference and its search zone are bundled together to copy to another region, the bundling add BO for FK Reference (which doesn’t include the Zone) ensures that the FK reference could be added before the zone without getting any validation errors.

## Owner Flags on Bundled Entities

The owner flag of the entities in an import bundle must match the owner flag of the target environment.

If you need to import objects that your source environment does not own, you must replace the owner flag in the import bundle with the owner flag of the target environment.

# Configuring Maintenance Objects for Bundling

Most base package meta-data objects are pre-configured to support bundling. All other objects must be manually configured.

If a base package maintenance object is pre-configured for bundling, the **Eligible For Bundling** option will be set to "Y" on the Options tab for the maintenance object.

To configure other objects for bundling, review the configuration tasks below and complete all those that apply:

| Configuration Task | Scope of Task |
|---|---|
| Make maintenance objects eligible for bundling | All objects to be included in the bundle. |
| Add a foreign key reference | All objects to be included in the bundle. |


Copyright © 2026, Oracle and/or its affiliates.

| Configuration Task | Scope of Task |
|---|---|
| Create a physical business object | All objects to be included in the bundle. |
| Create a bundling add business object | Only needed if there are objects with recursive or optional foreign key references. |
| Add the Current Bundle zone | Perform this step for any object where you want the Current Bundle zone to appear on the sidebar when on the maintenance page for this maintenance object. This is not required by the bundling process. |
| Create a custom Entity Search zone and add it to the Bundle Export portal | Perform this step if you would like the object to be listed in the Entity Search zone in the Bundle Export portal. This is not required by the bundling process. |

### Making Maintenance Objects Eligible for Bundling

The "Eligible For Bundling" maintenance object option must be set to "Y" for all bundled objects. Go to the Maintenance Object page and search for the maintenance object. On the Options tab, add a new option with the type **Eligible For Bundling**. Set the value to "Y" and click **Save**. This configuration must be in both the source environment and the target environment.

### Adding a Foreign Key Reference

Foreign key references are used for many purposes and most maintenance objects should have them anyway. Check the Maintenance Object options to see if there is an entry for the **Foreign Key Reference** option. If one is not configured, navigate to FK Reference and set up a foreign key reference for the primary table of the maintenance object. Configure that new FK reference on the maintenance object using an option.

### Creating a Physical Business Object

If the maintenance object does not already have a Physical BO configured, one must be created to support using it in a bundle. Refer to Creating a Physical Business Object for more information.

### Creating a Bundling Add Business Object

If the objects to be bundled have recursive foreign key references, optional foreign key references or child tables that include foreign key references, create a bundling add business object to avoid problems with referential integrity.

1. Navigate to Business Object and add a business object for the maintenance object. Do not fill in any of the UI oriented configuration.
2. The business object is created with a schema that looks like the physical structure of the maintenance object.
3. Remove all child tables and all elements that are not required.
4. Save the business object.
5. Navigate to Maintenance Object and search for the maintenance object you want to bundle.
6. On the **Option** tab, add a new option with the type **Bundling Add BO**. The value is the name of the bundling add business object you just created.


Copyright © 2026, Oracle and/or its affiliates.

### Adding the Current Bundle Zone

If you want the Current Bundle zone to appear on the sidebar when the a user is on the maintenance page for the maintenance object, you must add the Current Bundle zone as a context-sensitive zone for the maintenance page's navigation key.

1. Navigate to Context Sensitive Zone and search for the navigation key for the maintenance page of this object.
2. Add the Current Bundle zone F1-BNDLCTXT, to that navigation key.

### Adding a Customized Entity Search Query Zone to the Bundle Export Portal

If you want the maintenance object to be searchable in the Bundle Export portal in the Entity Search zone, you must first create an entity-specific query zone to search for the maintenance object. Then you must create a customized entity search zone that references this new query zone. Finally, you must add the customized entity search zone to the Bundle Export portal.

Note that not all base product entities that are eligible for bundling have a search configured for this zone. There is a limit to the number of searches that may be listed here. The base product has provided searches for only the most common objects that are bundled.

Create an entity-specific query zone to search for the maintenance object:

1. Navigate to Zone and search for one of the base query zones, such as the Algorithm Search zone F1-BNALGS.
2. Click the **Duplicate** button in the record actions section.
3. Enter a name and description for the new zone.
4. Click **Save**.
5. Customize the details of the parameters for your maintenance object.

Create a customized entity search zone. This step only needs to be done once. If you already have a customized search zone in the Bundle Export portal, skip this step.

1. Navigate to Zone and search for the F1-BNDLENTQ Entity Search zone.
2. Duplicate this zone.
3. Remove any references to base query zones.

Add the new entity-specific query zone to the customized entity search zone:

1. Locate the customized entity search zone for your Bundle Export portal. This is the zone created in the previous step.
2. Add the name of the query zone you created in the first step to the list of Query Zones.

Add the customized entity search zone to the Bundle Export portal. This step needs to be done only once.

1. Navigate to Portal and search for the Bundle Export portal, F1BNDLEM.
2. In the zone list, add the entity search zone you created above. (Add the new zone after the base entity search zone).
3. Save the portal.


Copyright © 2026, Oracle and/or its affiliates.

# Working with Bundles

Use the Bundle Export portal to create an export bundle. The export bundle contains a list of entities to be exported from the source environment. When you are ready to import the objects, use the Bundle Import portal to import the objects to the target environment.

- Creating Export Bundles
- Creating and Applying Import Bundles
- Editing Export Bundles
- Editing Import Bundles

## Creating Export Bundles

An export bundle contains a list of entities that can be imported into another environment.

To create an export bundle:

1. Log on to the source environment from which objects will be exported.
2. Select **Admin**, and then **Implementation Tools**, and then **Bundle Export**, and then **Add**.
3. Complete the fields in the Main section to define the bundle's basic properties.

> **Note**
>
> You can use the Entities section to add bundle entities now, or save the bundle and then add entities as described in step 5.

4. Click **Save** to exit the Edit dialog. The export bundle status is set to Pending.
5. While an export bundle is in Pending state, use any of the following methods to add entities to the bundle:
   a. Use the **Entity Search** zone on the Bundle Export portal to search for entities and add them to the bundle. If an entity is already in the bundle, you can remove it.
   b. To import entities from a .CSV file, click **Edit** on the Bundle Export portal, and then click **CSV File to Upload.** Specify the file name and location of the .CSV file containing the list of entities. Click **Submit** to upload the file, and then click **Save** to save the changes.
   c. Use the **Current Bundle** zone in the sidebar of the entity you want to add. (All entities that are configured to support bundling display a Current Bundle zone). This zone displays a list of all pending export bundles to which you can add the entity.
   d. When you check an entity into revision control, specify the export bundle on the **Revision Info** dialog.
6. When you have added all entities, click **Bundle** in the Bundle Actions zone on the Bundle Export portal. The export bundle state is set to Bundled and the Bundle Details zone displays the XML representation of every entity in the bundle.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The owner flags of the entities in the bundle must match the owner flag of the bundle itself. If the owner flags do not match, the system displays a warning message. Click **OK** to continue or **Cancel** to abort the bundle. If you click OK, you will need to resolve the owner flag discrepancy before you import the bundle to the target environment.

7. Copy the XML from the **Bundle Detail** zone to the clipboard (or to a text file). You can now create an import bundle and apply it to the target environment.

> **Note**
>
> If you need to make additional changes to the bundle, you must change the bundle state by selecting the **Back to Pending** button in the **Bundle Actions** zone.

## Creating and Applying Import Bundles

Import bundles define a group of entities to be added or updated in the target environment.

Before you create an import bundle, you must have already created an export bundle, added entities, and set the bundle's state to Bundled.

To create an import bundle and apply it to the target environment:

1. If you have not already copied the XML from the export bundle, do so now:
   a. Select **Admin**, and then **Implementation Tools**, and then **Bundle Export** and search for the bundle.
   b. Copy the XML from the **Bundle Detail** zone to the clipboard (or to a text file).
2. Log on to the target environment.
3. Select **Admin**, and then **Implementation Tools**, and then **Bundle Import**, and then **Add**.
4. In the **Bundle Actions** zone, click **Edit XML**.
5. Paste the contents of the clipboard (or text file if you created one) into the **Bundle Detail** zone.
6. Make any necessary changes to the XML and click **Save**. The status of the import bundle is set to Pending.

> **Note**
>
> Use caution when editing the XML to avoid validation errors.

7. To remove entities from the import bundle or change their sequence, click **Edit**. Enter your changes and click **Save** to exit the Edit dialog.
8. When you are ready to apply the bundle, click **Apply**. The import bundle state is set to Applied and the entities are added or updated in the target environment.


Copyright © 2026, Oracle and/or its affiliates.

## Editing Export Bundles

You can add or remove entities from an export bundle when it is in Pending state. You can also change the sequence of entities.

To edit to an export bundle that has already been bundled, you must change the bundle state by selecting the **Back to Pending** button on the Bundle Export portal.

To edit a pending export bundle:

1. Open the bundle in edit mode.
2. Click **Edit** on the Export Bundle portal.
3. Make any necessary changes on the edit dialog and then click **Save**.

## Editing Import Bundles

You can remove entities from an import when it is in Pending state. You can also change the sequence of entities and edit the generated XML.

To edit a pending import bundle:

1. Open the bundle in edit mode.
2. To edit the XML snapshot, click **Edit XML**. Edit the XML code as needed, then click **Save**.

> **Note**
>
> Use caution when editing the XML to avoid validation errors.

3. To remove entities or change their sequence, click **Edit**. Make any necessary changes and click **Save**.

# Revision Control

The topics in this section describe the revision control features in the application.

- About Revision Control
- Working with the Revision Control Zones
- Revision Control Portal

## About Revision Control

Revision control is a tool provided for the development phase of a project to allow a user to check out an object that is being worked on. In addition, it captures the version of the maintenance object when users check in an update, maintaining a history of the changes to the object.

If revision control is enabled for an object you must check out the object to change it. While the object is checked out no one else can work on it. You can revert all changes made since checking out an object, reinstate an older version of an object, recover a deleted object, and force a check in of an object if someone else has it checked out. You may also compare two versions of the object to more easily review changes.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Revision control does not keep your work separate from the environment. Because the metadata for maintenance objects is in the central database, any changes you make to an object while it is checked out will be visible to others and may impact their work.

Many of the maintenance objects used as configuration tools are already configured for revision control, but it is turned off by default. For example, business objects, algorithms, data areas, UI maps, and scripts are pre-configured for revision control.

- Turning On Revision Control
- Configuring Maintenance Objects for Revision Control

## Turning On Revision Control

Revision control is turned off by default for maintenance objects that are configured for revision control.

To turn on revision control:

1. Add the base package **Checked Out** zone to the Sidebar portal.
   a. Navigate to Portal.
   b. Search for the sidebar portal F1SDBTOOL.
   c. In the zone list for the portal, add the zone F1-USRCHKOUT.
2. Set up application security.

   For users to have access to revision control, they must belong to a user group that has access to the application service F1-OBJREVBOAS.
3. Add the revision control algorithm to the maintenance object that you want to have revision control.

   This step must be repeated for each maintenance object that you want to have revision control.
   a. Go to the Maintenance Object page and search for the maintenance object that you want to have revision control.
   b. On the **Algorithms** tab of the maintenance object, add the revision control algorithm F1-REVCTL.

## Configuring Maintenance Objects for Revision Control

Most configuration tool maintenance objects are pre-configured for revision control. You can configure other maintenance objects for revision control, as well.

To configure other objects for revision control:

1. Create a physical business object for the maintenance object, if it doesn't already have one. Refer to Creating a Physical Business Object for more information.
2. Add a foreign key reference to the maintenance object (if it doesn't already have one) so that the standard information string is displayed. Foreign key references are used for many purposes and most maintenance objects should have them anyway. Check the Maintenance Object options to see if there is an entry for the **Foreign Key Reference** option. If one is not configured, navigate to FK Reference and set up a foreign key


Copyright © 2026, Oracle and/or its affiliates.

reference for the primary table of the maintenance object. Configure that new FK reference on the maintenance object using an option.

3. Add the **Revision Control** zone to the maintenance object.
    a. Navigate to Context Sensitive Zone and search for the navigation key for the maintenance object.
    b. Add the Revision Control zone, F1-OBJREVCTL, to that navigation key.
4. Add the revision control algorithm to the maintenance object.
    a. Go to the Maintenance Object page and search for the maintenance object that you want to have revision control.
    b. On the **Algorithms** tab of the maintenance object, add the revision control algorithm F1-REVCTL.

# Working with the Revision Control Zones

You use two zones in the sidebar to work with revision controlled objects when revision control is turned on.

The **Revision Control** zone gives you several options for managing the revision of the currently displayed object. This zone also shows when the object was last revised and by whom. This information is linked to the **Revision History** portal which lists all of the versions of the object.

Using the Revision Control zone you can:

- Check out an object in order to change it.
- Check in an object so others will be able to work on it.
- Revert the object back to where it was at the last checkout.
- Force a check in of an object that is checked out by someone else. You need special access rights to force a check in.
- Compare the latest version of the object to the previous checked in version.
- Delete an object.

See the subsequent sections for more information about each action.

The **Checked Out** zone lists all of the objects that you currently have checked out. Clicking on an object listed in this zone will take you to the page for that object. The zone is collapsed if you have no objects checked out.

See Revision Control Portal for more information about Check In, Force Check In, and Check Out one or more records simultaneously.

### Checking Out an Object

If an entity's maintenance object is configured for revision control, you must check out the entity in order to change it. Go to the object you want to work on and click **Check Out** in the Revision Control sidebar zone.

The following points highlight behavior related to this functionality:

- When you create a record for a maintenance object that is managed with revision control, the system automatically associates the new record with a revision control in Checked Out status.


Copyright © 2026, Oracle and/or its affiliates.

- When you update an object governed by revision control, if you forget to check it out first, a dialog box informs you that the object is under revision control. You can select **OK** to check out the object and save your changes, or **Cancel** to stop the update.
- If another user has the object checked out, you will not be able to make changes.

> **Note**
>
> The system also supports Mass Check Out from the Revision Control search portal.

### Checking In an Object

Once you are finished with changes, you should check in the object. This captures the latest revision and allows others to check it out.

While viewing the object, click **Check In** in the Revision Control sidebar zone. You are prompted for an **External Reference** and **Detailed Description**. It is recommended to put meaningful information in both of these fields to allow for future users that review changes to this object have a good understanding of the reason for the change.

The **Keep Checked Out** option is useful if you are checking in a version of the object that is stable but allows you to keep it checked out for further changes. The latest revision is one available for you to restore to later if you want to back out newer changes.

If you are using Bundling to manage copying objects to another environment, use the Add to Bundle option to automatically include this object in a current open bundle.

> **Note**
>
> The system also supports Mass Check In from the Revision Control search portal.

### Reverting Changes

Reverting changes will undo any changes you made since you checked out an object.

While viewing the object, click **Revert** in the Revision Control sidebar zone. Once your change is reverted, the object is available to be checked out by another user.

> **Note**
>
> To restore the version of the object to a different revision than the most recent checked in revision, drill into the object's revision history. Refer to Revision Control Portal for more information.

> **Note**
>
> Any change to the object increments the version number. If you make changes and then revert your changes, internally the system updates the object to match the previously checked in revision. The version number, however, is incremented. It is not restored to the version number of the previous revision.


Copyright © 2026, Oracle and/or its affiliates.

### Comparing Versions

When viewing an object that is governed by revision control, a link on the Revision Control sidebar zone allows you to **Compare to Previous Version**. The comparison works as follows:

- When viewing an object that is currently checked out, clicking the link compares the current view of the object in the database with the snapshot view of the object from its most recent checked in revision.
- When viewing an object that is checked in, when clicking the link, the behavior differs based on whether or not the current view of the object matches the most recent revision.
  - If they match, you are shown the comparison between the most recent check in to the check in before that.
  - If they do not match, it means that some process changed the object outside of revisions. In that case, you are shown the difference between the current view of the object and the most recent revision.

> **Note**
>
> The revision history view of an object in the Revision Control Portal allows you to select any two revisions to compare.

### Forcing a Check In or Restore

If an object is checked out by another user and that person is not available to check it in or revert it, you may force a check in or force a restore (which is equivalent to a Revert action).

While viewing the object, click **Force Check In** in the Revision Control sidebar zone to keep the latest updates made by the checked out user. Click **Force Restore** in the Revision Control sidebar zone to discard the changes made by the checked out user and revert the object to the previously checked in revision.

> **Note**
>
> A user can only perform the Force Check In or Force Restore actions if they have security for the **Force Check In** and **Restore** access modes, respectively for the Object Revision BO application service (**F1-OBJREVBOAS**).

> **Note**
>
> The system also supports Mass Force Check In from the Revision Control search portal.

### Deleting an Object

If revision control is turned on for an object, you must use the **Revision Control** zone to delete it. The object must be checked in before it can be deleted.

While viewing the checked in object, click **Delete** in the Revision Control sidebar zone. You are prompted for an **External Reference** and **Detailed Description**. Provide this information and then click OK. The system creates a revision record before the object is deleted so that the deleted object can be restored.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the object is maintained on a legacy UI page, you remain on the page with a cleared input form. If the object is maintained on a portal page, after deleting an object, you remain on the maintenance portal page, which should now be blank, which is unusual. Unlike deleting an object using the record actions on the record's maintenance page, deleting from the sidebar zone lacks the metadata to bring the user to a better portal after deletion, such as the related query portal for the deleted object.

# Revision Control Portal

The **Revision History** portal lists the revision control history for a given object.

You can navigate to the **Revision History** portal from either a link in the **Revision Control** sidebar zone from that object's maintenance page or by going to **Revision Control** portal through the **Admin Menu** and using the search function to find the object whose history you want to view.

Use the broadcast icon to view the details of a given entry in the revision history.

To view the differences between two revision entries, click any two entries and then use the **Compare** button to see the differences.

You can restore a previous version of the object by clicking **Restore** in the row for the version that you want to restore. You would also use this option if an object was deleted and you want to reinstate it to a specific revision from before it was deleted.

See Working with Revision Control Zones for more information about tasks that can be performed through the Revision Control sidebar zone.

The following sections highlight other actions that are supported using the Revision Control Search portal besides searching for a specific revision. Navigate to this portal via **Admin Menu**, and then **Implementation Tools**, and then **Revision Control.**

### Mass Check In

To perform a mass check in of several objects that you have checked out, choose **Check In** from the Search By dropdown on Revision Control Search.

Use the filters to narrow down the list of objects that you have checked out, if desired.

Select the desired entries and click **Check In**.

### Mass Force Check In

If there are entities that are currently checked out by other users and they are not able to check them in, you can do a mass Force Check In of those objects. Choose **Force Check In** from the Search By dropdown.

Use the filters, including the Checked Out By User filter, to narrow down the list of objects to force check in on.

Select the desired entries and click **Force Check In**.

### Mass Check Out

If you plan to work on updates to several objects, you can opt to proactively check out the objects en masse. Choose **Check Out** from the Search By dropdown.


Copyright © 2026, Oracle and/or its affiliates.

Use the filters to narrow down the list of objects to check out. Note that only objects that are currently checked in are available to check out.

Select the desired entries and click **Check Out**.

# Information Lifecycle Management

Information Lifecycle Management (ILM) is designed to address data management issues, with a combination of processes and policies so that the appropriate solution can be applied to each phase of the data's lifecycle.

Data lifecycle typically refers to the fact that the most recent data is active in the system. As time progresses, the same data becomes old and unused. Older data becomes overhead to the application not only in terms of storage, but also in terms of performance. This older data's impact can be reduced by using advanced compression techniques, and can be put into slower and cheaper storage media. Depending on how often it's accessed, it can be removed from the system to make an overall savings of cost and performance. The target tables for ILM are transactional tables that have the potential to grow and become voluminous over time.


- The Approach to Implementing Information Lifecycle Management
- Enabling ILM for Supported Maintenance Objects
- Ongoing ILM Tasks
- Archived Foreign Keys


## The Approach to Implementing Information Lifecycle Management

This section describes the product approach to implementing ILM for its maintenance objects (MOs).

> **Note**
>
> The term archiving is used to cover any of the possible steps an implementation may take in their data management strategy, including compression, moving to cheaper storage, and removing the data altogether.

Age is the starting point of the ILM product implementation for some of its high volume data. In general "old" records are considered eligible to be archived. In the product solution, maintenance objects (MOs) that are enabled for ILM have an ILM Date on the primary table and the date is typically set to the record's creation date. (An MO may have special business rules for setting this date, in which case, a different date may be used to set the initial ILM Date). For implementations that want to use ILM to manage the records in the MO, the ILM date is used for defining partitions for the primary table.

There are cases where a record's age is not the only factor in determining whether or not it is eligible to be archived. There may be some MOs where an old record is still 'in progress' or 'active' and should not be archived. There may be other MOs where certain records should never be archived. To evaluate archive eligibility using information other than the ILM Date, the ILM enabled MOs include an ILM Archive switch that is used to explicitly mark records that have been evaluated and should be archived. This allows DBAs to monitor partitions based on age and the value of this switch to evaluate data that may be ready to be archived.

Evaluating records to determine their archive eligibility should still occur on "old" records. The expectation is that a large percentage of the old records will be eligible for archiving. The small



Copyright © 2026, Oracle and/or its affiliates.

number that may be ineligible could be updated with a more recent ILM date. This may cause the records to move into a different partition and can delay any further evaluation of those records until more time has passed.

For each MO enabled for ILM, the product provides a batch process to review "old" records and an ILM eligibility algorithm that contains business logic to evaluate the record and mark it eligible for archiving or not. The following sections provide more information about the batch process and algorithm functionality.

- Batch Processes
- Eligibility Algorithm

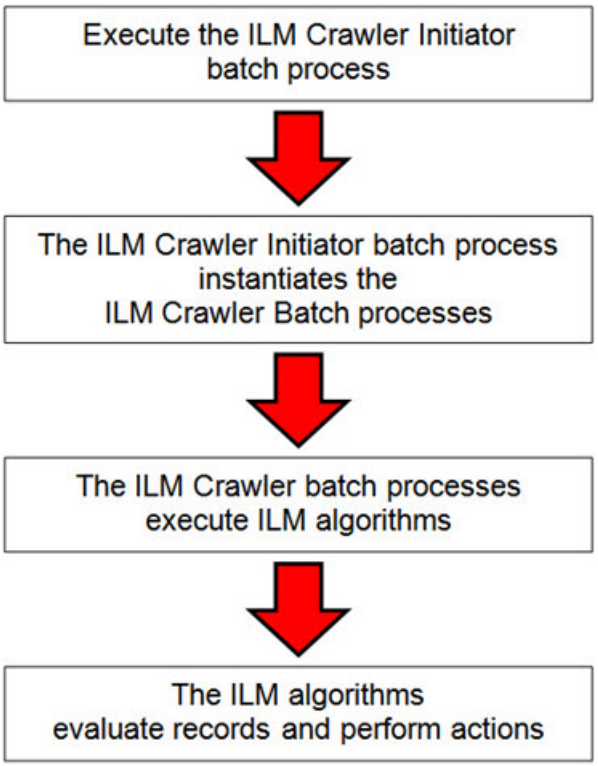
## Batch Processes

There are two main types of batch processes that manage data for ILM in the application: ILM Crawler Initiator and individual ILM Crawlers (one for each MO that is configured for ILM).

- **ILM Crawler Initiator**: (`F1-ILMIN`) - The ILM Crawler Initiator is a *driver* batch process that starts the individual ILM Crawler batch control as defined by the MO's options.

  **Restartable**: In case of server failure, the ILM Crawler Initiator process can be restarted, which will also restart the ILM Crawler processes.

- **ILM Crawler**: Each maintenance object that is configured for ILM defines an ILM Crawler. These are *child* batch processes that can be started either by the ILM Crawler Initiator or by a standalone batch submission.




The ILM Crawler batch process selects records whose retention period has elapsed and invokes the MO's ILM eligibility algorithm to determine if the record is ready to be archived or not. The ILM eligibility algorithm is responsible for setting the record's ILM archive switch to 'Y' and updating the ILM date, if necessary.


Copyright © 2026, Oracle and/or its affiliates.

The retention period defines the period that records are considered active. It spans the system date and cutoff date (calculated as system date - retention days).

The retention days of an MO is derived as follows:

- If the ILM Retention Days MO option is defined, that is used.
- Otherwise, the Default Retention Days from the ILM Master Configuration record is used.

An error is issued if no retention period is found.

The crawler calculates the cutoff date and selects all records whose ILM archive switch is 'N' and whose ILM date is prior to the cutoff date. Each record returned is subject to ILM eligibility.

If the Override Cutoff Date parameter is supplied, it will be used instead of the calculated cutoff date. An error is issued if the override cutoff date is later than the calculated cutoff date. This parameter is useful if an object has many years of historic data eligible for archiving. Setting this parameter allows for widening the retention period and therefore limiting the process to a shorter period for initial processing.

> **Note**
>
> ILM Crawler batch processes are designed not to interfere with current online or batch processing. Because of this, these batch processes can run throughout the day.

> **Note**
>
> Before passing the cut-off date to the algorithm, the ILM crawler ensures that the number of days calculated (System Date – override cut-off date) is more than the retention period specified in the MO option or the Master Configuration. If the number of days calculated **is less than** the retention period specified on the MO option or the Master Configuration, then it throws an error.

## Eligibility Algorithm

Algorithms are triggered by the ILM batch crawler for the maintenance object. The key responsibility of the ILM algorithm is to determine whether a record can be marked as ready to be archived or not. If a record is determined to be ready for archive, the algorithm should set the ILM Archive switch to Y. If not, the algorithm leaves the switch set to N and may decide to update the ILM Date to something more recent (like the System Date) to ensure that the record does not get evaluated again until the future.

This algorithm is plugged into the Maintenance Object — Algorithm collection.

Oracle Utilities Application Framework provides the algorithm **ILM Eligibility Based on Status** (`F1-ILMELIG`) to support the ILM batch crawler. Refer to the algorithm type description for details about how this algorithm works. If a maintenance object has special business rules that are evaluated to determine the eligibility for ILM, a custom algorithm can be created and applied by the implementation team.

# Enabling ILM for Supported Maintenance Objects

In order to enable ILM for one or more maintenance objects, several steps are needed in both the configuration and in the database. This section describes some configuration enabled by


Copyright © 2026, Oracle and/or its affiliates.

default and some steps that must be taken in order to fully implement ILM. While some configuration is enabled by default, it will not be used unless ILM is fully configured.

Managing a maintenance object's data for ILM involves the following aspects:

- Determining which records are eligible for archiving and marking them as such. Depending on the entity, this step is implemented using one of this methods:
  - **Configurable.** This option relies on a configurable eligibility algorithm that is processed in batch periodically.
  - **Non-configurable batch.** This option uses a specific batch process that embeds non-configurable eligibility rules as part of its logic for performance reasons.
  - **Set at creation time.** Records in very high volume tables that rarely change are marked as eligible for archival at creation time for performance reasons. For example, measurement data, some snapshot tables, etc.
- Partitioning the entity's tables and archiving old partitions as needed.

Each maintenance object that follows the configurable option has the following provided out of the box:

- **Special Table Columns**: Maintenance objects that support ILM include two specific columns: ILM Archive Switch (`ILM_ARCH_SW`) and ILM Date (`ILM_DT`).
- **Crawler Batch Process**: A "crawler" batch process is provided for each maintenance object that supports ILM and it is plugged into the MO as an option. Refer to Batch Processes for more information.
- **ILM Eligibility Algorithm**: Each maintenance object that is configured for ILM defines an eligibility algorithm that executes the logic to set the ILM Archive switch appropriately. This is plugged in to the MO algorithm collection.

If an implementation decides to implement ILM, there are steps that need to be followed, which are highlighted below.

### Create the Master Configuration Record

The first step when enabling ILM is to create the **ILM Configuration** master configuration record.

The master configuration for ILM Configuration defines global parameters for all ILM eligible maintenance objects. For example, the Default Retention Period. In addition, your product may implement additional configuration. Refer to the embedded help for specific details about the information supported for your product's ILM configuration.

In addition, the user interface for this master configuration record displays summary information about all the maintenance objects that are configured to use ILM.

### Confirm the Maintenance Objects to Enable

In viewing the list of maintenance objects that support ILM in the ILM master configuration page, your implementation may choose to enable ILM for only a subset of the supported maintenance objects. For example, some of the maintenance objects may not be relevant for your implementation. Or perhaps, the functionality provided by the maintenance object is used, but your implementation does not expect a high volume of data.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Customizing the list of maintenance objects managed by ILM is not supported in a cloud installation.

For each maintenance object that your implementation has confirmed for ILM, the following steps should be taken:

- Determine if the maintenance object should have a different number of default retention days than the system wide value defined on the master configuration. If so, use the MO option **ILM Retention Period in Days** to enter an override option for this maintenance object.
- When applicable, review the functionality of the ILM Eligibility algorithm provided by the product for this maintenance object. Each algorithm may support additional configuration based on business needs. If your organization has special business rules that aren't satisfied by the algorithm provided by the product, a custom algorithm may be provided to override the base algorithm.

For each maintenance object that follows the configurable option and that your implementation does not want to enable for ILM, inactivate the eligibility algorithm. This will ensure that the ILM Crawler Initiator background process does not submit the crawler batch job for the maintenance object in question.

- Go to the Maintenance Object - Algorithm tab for each maintenance object and inactivate the **ILM Eligibility** algorithm.

### Database Administrator Tasks

In order to implement ILM for each of the maintenance objects determined above, your database administrator must perform several steps in the database for the tables related to each MO. The following points are a summary of those steps. More detail can be found in the Information Lifecycle Management section of your product's *Database Administration Guide*.

- **Initializing ILM Date**: Your existing tables that are enabled for ILM may not have the ILM Date and ILM Archive switch initialized on all existing records. When choosing to enable ILM, a first step is to initialize this data based on recommendations provided in the *Database Administration Guide*.
- **Referential Integrity**: The recommended partitioning strategy for child tables in a maintenance object is referential partitioning. In order to implement this, database referential integrity features must be enabled.
- **Partitioning**: This provides a way in which the data can segregate into multiple table partitions and will help in better management of the lifecycle of the data.

### Schedule the ILM Crawler Initiator

The final step of enabling the system for ILM is to schedule the ILM crawler initiator `F1-ILMIN` regularly based on your implementation's need. It is recommended to only schedule this batch process once all the required database activities are complete.

## Ongoing ILM Tasks

For an environment where ILM is enabled, besides the periodic execution of the ILM crawler batch processes to review and mark records, your database administrator has ongoing tasks.


Copyright © 2026, Oracle and/or its affiliates.

The DBA reviews and maintains partitions and identifies partitions that may warrant some type of archiving step. The Information Lifecycle Management section of your product's *Database Administration Guide* provides more information for your DBA.

## Archived Foreign Keys

If your DBA chooses to archive a partition, there may be records in the system that refer to one of the archived records as a foreign key.

When a user attempts to view a record using a hyperlink or drill down mechanism, if the implementation of the navigation uses FK Reference functionality, the system will first check if the record exists. If not, it will display a message to the user indicating that the record has been archived.

# Configuration Tools

This section describes tools to facilitate detailed business configuration. The configuration tools allow you to extend both the front-end user interface as well as create and define specialized back-end services.

- Business Objects
- Data Areas
- Business Services
- User Interface (UI) Options
- Process Flows
- Action Method
- Algorithms
- Script Options
- Maintaining Managed Content
- Defining Entity Tags
- Advanced Schema Topics
- Business Event Log
- Miscellaneous Topics
- Extensions Dashboard

## Business Objects

A maintenance object defines the physical tables that are used to capture all the possible details for an entity in the system. A business object is tool provided to further define business rules for a maintenance object.

This section provides an overview of business objects and describes how to maintain them.

- Understanding Business Objects
- Defining Business Objects
- Advanced BO Tips and Techniques


Copyright © 2026, Oracle and/or its affiliates.

- Defining Status Reasons

# Understanding Business Objects

The topics in this section describe background topics relevant to business objects.

- What Is A Business Object?
- Business Object Inheritance
- Each Business Object Can Have A Different Lifecycle
- BO Algorithm Execution Summary
- Granting Access To Business Objects

## What Is A Business Object?

A business object (BO) is a powerful tool used throughout the system. For many maintenance objects, a BO is a key attribute of the record used to define the data it captures, its user interface behavior and its business rules. Some business objects support the definition of a lifecycle, capturing different states that a record may go through, allowing for different business rules to be executed along the way. This type of business object is considered the "identifying" or "governing" business object. We will see later that other types of BOs exist that are different from the "identifying" business object.

The use of business objects allows for extensibility and customization of product delivered maintenance objects. There are many options to adjust the behavior of base delivered business objects. In addition, implementations may introduce their own business objects if the base product delivered objects do not meet their business needs.

> **Note**
>
> Not all maintenance objects in the product support business objects as a "identifying" or "governing" tool. This is the standard going forward for new maintenance objects. However, there are some maintenance objects created before this became a standard.

- A Business Object Has a Schema
- A Business Object May Define Business Rules
- A Business Object Defines its User Interface
- A Business Object Defines Options
- Invoking A Business Object
- Determine the Identifying BO
- Base Business Objects

### A Business Object Has a Schema

A business object has elements. The elements are a logical view of fields and columns in one of the maintenance object's tables. The structure of a business object is defined using an XML schema. The main purpose of the schema is to identify all the elements from the maintenance object that are included in the business object and map them to the corresponding maintenance object fields. Every element in the BO schema must be stored somewhere in the maintenance object. The BO may not define elements that are derived.


Copyright © 2026, Oracle and/or its affiliates.

When defining elements for the primary table or the language table (for an administrative object) there is no need to define the name of the physical table in the schema. The system infers this information. The following is a snippet of a schema:

```
<schema>
    <migrationPlan mapField="MIGR_PLAN_CD" suppress="true" isPrimeKey="true"/>
    <bo mapField="BUS_OBJ_CD" fkRef="F1-BUSOB"/>
    <customizationOwner mapField="OWNER_FLG" suppress="input"/>
    <version mapField="VERSION" suppress="true"/>
    <description mapField="DESCR"/>
    <longDescription mapField="DESCRLONG"/>
  â€¦
```

Many maintenance objects have child table collections (e.g., a collection of names for a person, or a collection of persons on an account). Depending on the requirements, the business object may define the full collection such that the user will maintain the information in a grid. However, the schema also supports "flattening" records in a child table so that they can be treated as if they were singular elements. The following are examples of each:

**Example** of a child table. This is a snippet of the Instructions collection on the migration plan business object. You can see that the list attribute defines the child table and all elements within it map to the appropriate column in that table.

```
   <migrationPlanInstruction type="list" mapChild="F1_MIGR_PLAN_INSTR">
         <migrationPlan mapField="MIGR_PLAN_CD" suppress="true"/>
         <sequence mapField="PLAN_INSTR_SEQ" suppress="true"/>
         <instructionSequence mapField="INSTR_SEQ"/>
         <instructionType mapField="INSTR_TYPE_FLG"/>
         <parentInstructionSequence mapField="PARENT_INSTR_SEQ"/>
         <businessObject mapField="BUS_OBJ_CD" fkRef="F1-BOMO"/>
...
```

**Example** of a simple "flattened" field. The business object for Status Reason includes an element called Usage, which maps to a pre-defined characteristic of type **F1-SRUSG**. The "row" defines which child table is being flattened and the attributes of the row in that child that uniquely identify it.

```
    <usage mdField="STATUS_RSN_USAGE" mapField="CHAR_VAL">
        <row mapChild="F1_BUS_OBJ_STATUS_RSN_CHAR">
            <CHAR_TYPE_CD is="F1-SRUSG"/>
            <SEQ_NUM is="1"/>
        </row>
    </usage>
```

**Example** of a "flattened row". This business object for Account includes a single row for the Person collection where only the "financially responsible, main" customer is defined. The "accountPerson" attribute defines one field from that row (the Person Id) and includes the "˜flattening" criteria in the "row" information. In addition, a second field from that same row ("accountRelType") is defined. Instead of having to repeat the flattening criteria, the "rowRef" attribute identifies the element that includes the flattening.

```
    <accountPerson mapField="PER_ID">
        <row mapChild="CI_ACCT_PER">
            <MAIN_CUST_SW is="true"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <FIN_RESP_SW default="true"/>
          </row>
     </accountPerson>
     <accountRelType mapField="ACCT_REL_TYPE_CD" rowRef="accountPerson"
dataType="string"/>
```

**Example** of a "flattened list". The business object for Tax Bill Type includes an list of valid algorithms for "bill completion". The Sequence and the Algorithm are presented in a list. The list element identifies the child table and the "˜rowFilter" identifies the information about the list that is common.

```
       <taxBillCompletion type="list" mapChild="C1_TAX_BILL_TYPE_ALG">
         <rowFilter suppress="true" private="true">
             <TAX_BILL_TYPE_SEVT_FLG is="C1BC"/>
         </rowFilter>
         <sequence mapField="SEQ_NUM"/>
         <algorithm mapField="ALG_CD" fkRef="F1-ALG"/>
     </taxBillCompletion>
```

In addition, many maintenance objects support an XML structure field within the entity. These fields may be of data type CLOB or XML. One or more business object elements may be mapped to the MO's XML structure field. These elements may be defined in different logical places in the business object schema based on what makes sense for the business rules. When updating the MO, the system builds a type of XML document that includes all the elements mapped to the XML structure and stores it in one column. The following is an example of elements mapped to an XML column:

```
        <filePath mdField="F1_FILE_PATH" mapXML="MST_CONFIG_DATA"
required="true"/>
        <characterEncoding mdField="F1_CHAR_ENCODING"
mapXML="MST_CONFIG_DATA"/>
```

> **Note**
>
> If the MO's XML structure field is of the data type XML, the database will allow searching for records based on that data, assuming appropriate indexes are defined. If the MO's XML structure field is of the data type CLOB, indexing or joining to elements in this column via an SQL statement is not typically supported. Note that most MOs are currently using the CLOB data type for the XML structure column, if provided.

Some business objects may have child tables that allow data to be stored in an XML structure field. The schema language supports defining elements from those fields in your schema as well.

Besides including information about the physical "mapping" of the element to its appropriate table / field location in the maintenance object, the schema supports additional syntax to provide the ability to define basic validation and data manipulation rules, including:

- Identifying the primary key of the record or the primary key of the a row in a list.
- Identifying which elements are required when adding or changing a record.
- Default values when no data is supplied on an Add.


Copyright © 2026, Oracle and/or its affiliates.

- For elements that are lookup values, the lookup may be specified to validate that the value of the element is a valid lookup value.
- For elements that are foreign keys to another table, the FK Reference may be specified to validate the data.

The system will check the validity of the data based on the schema definition obviating the need for any special algorithm to check this validation.

In addition, the schema language may include some attributes that are used to auto-render the view of the record on the user interface, such as the `suppress` attribute. Refer to BO Defines its User Interface for more information.

> **Note**
>
> Refer to Schema Syntax for the complete list of the XML nodes and attributes available to you when you construct a schema.

A business object's schema may include a subset of the fields and tables defined in the maintenance object. There are two reasons for this:

- The fields or tables may not be applicable to the type of record the business object is governing. For example, a field that is specific to gas may not be included on a Device business object that is specific to electric meters.
- The information is not maintained through the business object, but rather maintained separately. For example, many BO based maintenance objects include a Log table. The records in the log table are typically not included the BO because they are viewed and maintained separately from the business object.

## A Business Object May Define Business Rules

A business object may define business rules that govern the behavior of entities of this type. The following table describes supported system events that can be associated with such rules.


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
| --- | --- | --- |
| **Audit** | Optional | Algorithms of this type may be used to audit certain changes made to business object instances.<br>The system hands over to the algorithms a summary of all the elements that were changed throughout a specific call to update an object. For each element its original value before the change as well as its new value are provided.<br>It is the responsibility of such algorithms to log the changes if and where appropriate, for example as a log entry in the maintenance object's log table or an entry in the business event log.<br>These plug-ins are executed after all the post-processing rules are executed.<br>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.<br>By default all elements of the business object are subject to auditing. You can however mark certain elements to be excluded from the auditing process using the **noAudit** schema attribute. Marking an element as not auditable will prevent it from ever appearing as a changed element in the business object's audit plug-in spot. In addition, if the only elements that changed in a BO are ones marked to not audit, the audit algorithm is not even called. Refer to Schema Syntax for more information on this attribute. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
| --- | --- | --- |
| **Information** | Optional | We use the term "Business Object Information" to describe the basic information that appears throughout the system to describe an entity defined by the business object. The data that appears in this information description is constructed using this algorithm.<br>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number found on the business object closest to the current business object in the inheritance hierarchy. Refer to Business Object inheritance for more information. |
| `Post-Processing` | Optional | Algorithms of this type may be used to perform additional business logic after a business object instance has been processed.<br>For example, such an algorithm may be used to create a To Do Entry or add a log record as part of the business object logical transaction. These plug-ins are executed after all the validation rules are executed.<br>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| `Pre-Processing` | Optional | Algorithms of this type further populates a request to maintain a business object instance right before it is processed.<br>Such algorithm may be used to "massage" a business object's elements prior to any maintenance object processing. For example, although simple element-level defaulting is supported by schema attributes. you may use this type of algorithm to default element values that are more sophisticated.<br>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information. |
| **Validation** | Optional | Algorithms of this type may be used to validate a business object instance when added, updated or deleted.<br>Simple element-level validation is supported by schema attributes. Note that element-level validation is executed before any maintenance object processing. For more sophisticated rules you create validation algorithms and associate them with your business object. BO validation algorithms are only executed after "core validation" held in the MO is passed.<br>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information. |

The system applies business object rules (schema based and algorithms) whenever a business object instance is added, changed or deleted. This is only possible when the call is made via the maintenance object service. For example, when made via business object interaction ("invoke BO"), the MO's maintenance page or inbound web services that reference the BO. In addition the system must be able to determine the identifying business object associated with the actual object being processed. If the business object cannot be determined for a maintenance object instance business object rules are not applied.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Pre-Processing is special.** The pre-processing algorithm plug-in spot is unique in that it only applies during a BO interaction. It is executed prior to any maintenance object processing. It means that when performing add, change or delete via the maintenance object service, the pre-processing plug-in is not executed.

> **Caution**
>
> Direct entity updates bypass business rules! As mentioned above, it is the maintenance object service layer that applies business object rules. Processes that directly update entities not via the maintenance object service bypass any business object rules you may have configured.

> **NOT_SUPPORTED**
>
> Refer to BO Algorithm Execution Order for a summary of when these algorithms are executed with respect to lifecycle algorithms.

The plug-in spots described above are available for all business objects and they are executed by the system when processing adds or updates to the business object. It is possible for a specific maintenance object to define a special plug-in spot for business objects of that MO. When this happens, the maintenance object identifies the special algorithm entity lookup value as an MO option: **Valid BO System Event**, causing the BO Algorithm collection to include that system event in its list.

## A Business Object Defines its User Interface

One of the responsibilities of an identifying business object is to define its user interface rules for viewing and maintenance of its record. The standard implementation for maintaining a business object is that a maintenance portal is used to display a record. This portal includes a "map" zone that displays the information about the business object. To add or make changes to a record, the user clicks a button that launches a maintenance BPA script which displays a maintenance "map".

The display and maintenance "maps" are driven by the business object.

- **Schema based UI.** This is the preferred option. The BO's schema definition and UI Hints define user interface behavior that results in generated HTML. The schema language includes some basic display attributes such as **label** and **suppress**. UI hints provide many additional tags and elements that allow dynamic generation of formatted UI Maps. For more complex behavior in the user interface, for example where javascript is needed, UI map fragments may be defined within the schema via UI hints. In this way only complex UI behavior warrants small snippets of javascript and HTML. However the rendering of standard fields can be dynamically rendered. UI map fragments also allow for derived fields to be included in the user interface.
- **Explicit UI map.** The BO may rarely define a full UI map where all the information is displayed based on the map's HTML. Note that for a child BO, the maps may be inherited by a parent BO (or any BO "up the chain"). This technique is less flexible for describing a business object and as such is mainly supported for backward compatibility.


Copyright © 2026, Oracle and/or its affiliates.

A business object schema may include data areas for segments of its schema definition to allow for reusable components. In this case the data area would also include schema attributes and UI hints for the elements that it is including.

> **Note**
>
> Refer to UI Hint Syntax for detailed information about the supported syntax.

As mentioned in Business Object Inheritance, schemas are not inherited on a child business object. As such, when using UI hints for automatic UI rendering, the child BO must define the full schema with all the definitions. A good business object hierarchy will be designed for reuse meaning that the child BO will include the parent BO schema or alternatively, the BO schemas will include reusable data areas.

By default, the input form used to add or update an entity is presented in the large drawer panel. If the small or medium drawer panel fits better to display this form, you can configure the **Maintenance Drawer Size** BO option accordingly.

## A Business Object Defines Options

A business object may be associated with configuration options that control specific features. For example, the scripts used to maintain and display a business object are defined as options. Refer to the detailed description of each option type for more information on the functionality it controls.

> **Note**
>
> **You can add new options types**. Your implementation may want to add additional option types. For example, your implementation may have plug-in driven logic that would benefit from a new option. To do that, add your new values to the customizable lookup field `BUS_OBJ_OPT_FLG`. If you add a new option type for a business option, you must update its maintenance object to declare this new option type. Otherwise, it won't appear on the option type dropdown. You do that by referencing the new option type as a **Valid BO Option Type** maintenance object option.

Some options may be defined for a specific lifecycle status. For example, marking a status specific base algorithm as inactive is defined as an option on that staus definition record.

> **Note**
>
> **You can add new options types**. Your implementation may want to add additional option types. For example, your implementation may have plug-in driven logic that would benefit from a new option. To do that, add your new values to the customizable lookup field `BO_OPT_FLG`. If you add a new option type for a status, you must update the business object's maintenance object to declare this new option type. Otherwise, it won't appear on the option type dropdown. You do that by referencing the new option type as a **Valid BO Status Option Type** maintenance object option.


Copyright © 2026, Oracle and/or its affiliates.

## Invoking A Business Object

We have talked about defining a business object. This section describes how business objects are used throughout the system to view, add and update records.

- Various parameters for zones that are used to display data in the system include support for retrieving data by referencing a business object. The zone code will "invoke" the BO, meaning that the record will be retrieved using the referenced BO.
- The system's scripting language includes a step type to "invoke BO". This allows for BPA scripts, service scripts and plug-in scripts to retrieve information and add or update records using BO interaction.
- Inbound web services may reference a business object in its operations collection. This allows external systems to add or update records in our product via web service interaction.

Often when configuring a zone or writing a script, the BO to use in the "invoke BO" statement should be the identifying BO of the record. As such, often the script will include steps prior to the "invoke BO" step to "determine the identifying BO of the record" and once the identifying BO is found, the script step will invoke that BO. Note that zones and inbound web services reference a BO directly. In each case, if the BO to use should be dynamic, then the zone / inbound web service should reference a service script that can perform the steps to identify the BO and then invoke that BO.

It should be noted however that the BO used in an "invoke BO" statement (or referenced in an inbound web service) **does not have to match** the identifying BO for the record. Here are some examples of where this may be true:

- A script may only require a subset of elements for a record and not the entire record. In this case, it is better for performance purposes to define a special BO (sometimes called a "lite" BO or a "mini" BO) that only defines the needed elements. When the system retrieves the data, it will only access the tables that are included in the BO's schema definition. In addition, if there are no elements that map to an XML structure field, the system will skip any parsing of that column. Similarly, if a script is **updating** a subset of elements on a record, it may be beneficial to use a "mini" BO to do the updates.

  > **Note**
  >
  > Please note the following with respect to using a mini BO. This BO is only used for its schema. This type of BO would not define algorithms or a lifecycle. Because the BO is special, it often should not be able to be used as any record's identifying BO. To control that, these BOs are often configured to not allow new instances. Refer to Determine the Identifying BO for more information.

- The maintenance object to be added or updated in a script may not support business objects as "identifying BOs". For example, Batch Control maintenance object does not have an identifying BO. However, scripts may still wish to retrieve data (or make updates) to these types of records. An easy way to achieve that goal is to define a business object and use "invoke BO" to access the data.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Not all maintenance objects support being maintained through a business object interaction. This is true in a small number of older objects where the underlying maintenance service includes additional functionality besides simply updating the database tables. These maintenance objects are identified via the MO option **BO Maintenance**, set to `N`.

- Some functionality may be trying to add or update records for a maintenance object in a 'physical' manner and do not want or need to use the object's identifying BO. Or the MO may not have an identifying BO. For example, revision control takes a snapshot of a record for audit purpose and to be able to restore a previous version. In this case, the system wants to capture a full "physical" view of the record. To do this, a special "physical" BO may be created that includes all (or most of) the columns and the child tables.

> **Note**
>
> Like the mini BO, the physical BO would not define algorithms or a lifecycle and should not be able to be used as any record's identifying BO. To control that, these BOs are often configured to not allow new instances. Refer to Determine the Identifying BO for more information.

> **Note**
>
> To reiterate, the BO referenced in the "invoke BO" statement or referenced in an inbound web service does not have to match the identifying BO and does not have to be configured to "allow new instances".

## Determine the Identifying BO

As mentioned in other topics, the identifying BO is the business object that governs the business rules for a record. This is the business object that the record will be validated against when any additions or changes are made to the record as long as updates are made via the maintenance service. This includes using "invoke BO" for add or update, using inbound web service interaction and for access to the maintenance page service (via an old style fixed page or via a business service).

How does the system determine the identifying BO? An algorithm plugged into the maintenance object (the **Determine BO** plug-in spot) is responsible for this. If the maintenance object is not configured with an algorithm for this plug-in spot, or no BO is found by the algorithm, no BO business rules are applied.

Most maintenance objects in the system capture the record's identifying BO directly on the record. However, it is possible to define the identifying BO somewhere else. For example, there may be some maintenance objects that are master or transaction objects with an associated "type" object where the identifying BO is defined on its "type" object. Note that the standard Determine BO algorithm plugged into most maintenance objects (**F1-STD-DTMBO - Determine Standard Business Object**) checks for these two conditions.

There may also be cases where a single identifying BO is used for all BOs for a given MO. This may be an option used for some older maintenance object created prior to the business object functionality when implementations wish to introduce custom business rules that are


Copyright © 2026, Oracle and/or its affiliates.

common for all records of that MO. The product provides a base algorithm type (**F1–MOBO - Determine Specific Business Object**) that captures the BO as a parameter.

## Base Business Objects

For each maintenance object (MO) that supports an "identifying" business object, the type of business object provided by the product depends on the functionality and expected use by implementations. The following are some common patterns.

- There are MOs where the product provides base BOs that implementations may use if applicable for their business rules. In addition, it is expected that implementation will define custom BOs to support their business needs. Good examples of this type of MO are any of the various "rule" MOs. For example, calculation rule in Oracle Utilities Customer Care and Billing or the usage rule in Oracle Utilities Meter Data Management. The product provides business objects for common rules but each implementation could have special rules that they need to implement and will need to create custom business objects.
- There are MOs where the product provides base BOs that supply common behavior for an object. Implementations may find that supplied the business objects match their business requirements and use the BOs as is. It is expected, however that for many implementations, their business rules will require additional elements to be captured or have special rules to apply. In this case the base business objects may be extended. This scenario may apply to 'master' data objects in various products such as the Device or Meter.
- There are MOs where the product may deliver a base BO that is not expected to satisfy most implementations because different jurisdictions or different implementations will typically have their own rules. In this case the base delivered BO can be used as a template or starting point for custom defined BOs. Some examples of this are Rebate Claim in Oracle Utilities Customer Care and Billing.
- There are MOs where the expectation is that every implementation will have different requirements for the type of data to capture and the product will not supply base BOs that can be used as the "identifying" BO. However, it may supply a "parent" BO that defines the lifecycle and many of the business rules that it expects all records to follow. In these scenarios, the implementations will create "child" BOs that will serve as the "identifying" BOs and refer to the base "parent" BO for many of its rules through inheritance. An example of this is the Activity in Oracle Utilities Mobile Workforce Management.
- There are some scenarios where the base product provides business objects and the expectation is that implementations will use the business objects as delivered with little or no customization. This is a case where the system used business objects to implement product functionality, not because there is an expectation that the implementers will extend the functionality, but because the business object model is the favored development tool even for the product. The objects delivered for Content Migration Assistant are an example.

> **Note**
>
> Not all maintenance objects in the product support business objects as a "identifying" or "governing" tool. This is the standard going forward for new maintenance objects. However, there are some maintenance objects created before this became a standard.

For all maintenance objects, the base product may provide additional BOs that are not meant to be "identifying" BOs, but instead are provided to support functionality to interact with the MO using the BO as a tool as described in Invoking a BO.

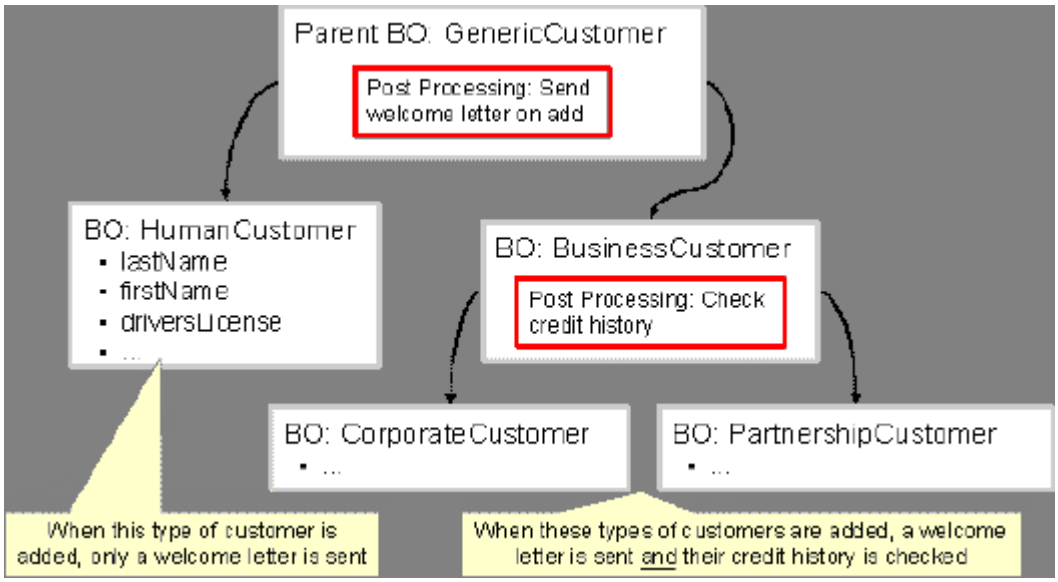

Copyright © 2026, Oracle and/or its affiliates.

- One or more "mini" or "lite" BOs may be supplied for a maintenance object. This may be found when the product has functionality to retrieve a subset of elements for the maintenance object via scripting or via a user interface.
- A "physical" BO may be supplied. This a BO that typically includes all tables and all fields of the maintenance object in there "physical" form. In other words, there is no "flattening" of child tables and any XML structure fields are defined as a single field. Physical BOs are used in system processing where the full record needs to be captured as is. Some functionality that uses a physical BO includes bundling, revision control and the pre-compare algorithm for CMA to adjust data prior to comparing. If there are any maintenance objects that do not have a base supplied physical BO, refer to Creating a Physical Business Object for steps on how to provide one.
- A "bundling add" BO may be supplied. Refer to Recursive Key References for more information as to why this type of BO may be supplied.

## Business Object Inheritance

A business object may inherit business rules from another business object by referencing the latter as its parent. A child business object can also have children, and so on. A parent's rules automatically apply to all of its children (no compilation - it's immediate). A child business object can always introduce rules of its own but never remove or bypass an inherited rule.

The following is an illustration of multiple levels of business object inheritance.




Notice how the "Business Customer" business object extends its parent rules to also enforce a credit history check on all types of customers associated with its child business objects.

Most types of business object system events allows for multiple algorithms to be executed. For example, you can have multiple **Validation** algorithms defined for a business object. For these, the system executes all algorithms at all levels in the inheritance chain starting from the highest-level parent business object moving on to lower levels.

Other types of system events allows for a single algorithm to be executed. For example, you can only have one **Information** algorithm to format the standard description of a business object instance. For these, the system executes the one at the level nearest to the business object currently being processed.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The parent and its children must reference the same maintenance object.

> **Note**
>
> **Data structures are not inherited.** While you can declare schemas on parent business objects, their children will not inherit them. A good practice is to design child business object schemas to **include** their parent business object's schema.

> **Note**
>
> **User interface maps are inherited.** When determining if the business object has a UI map to use for UI rendering, the system looks for display and maintenance maps linked to the BO as options. If the identifying BO does not have maps defined, the system follows “up the chain” of inheritance until it finds a map to use. This allows for a child BO to be used to extend business rules of a parent BO but inherit its user interface behavior. Refer to Business Object Defines its User Interface for more information.

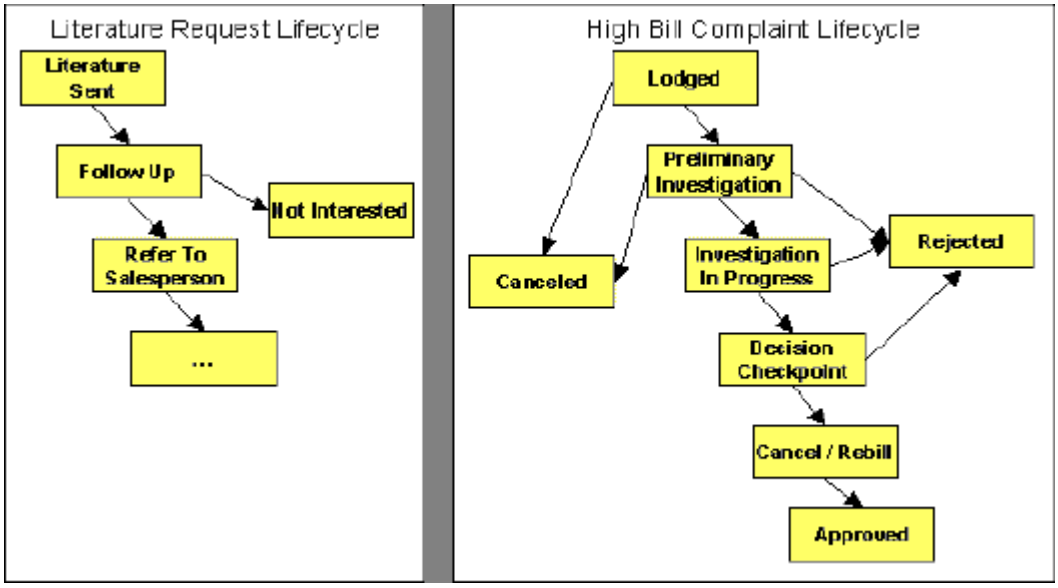
> **Note**
>
> **Use Inheritance Wisely.** While it is intellectually attractive to abstract behavior into parent BOs to avoid redundant logic and simplify maintenance, before doing this weigh the reuse benefits against the cost in transparency, as it is not easy to maintain a complex hierarchy of business objects.

## Each Business Object Can Have A Different Lifecycle

Many maintenance objects have a status column that holds the business entity's current state within its lifecycle. Rules that govern lifecycle state transition (what is its initial state, when can it transition to another state, and so on.) and the behavior associated with each state are referred to as lifecycle rules. Older Maintenance Objects, such as To Do Entry, have predefined lifecycles whose rules are governed by the base-package and cannot be changed. The lifecycle of newer Maintenance Objects exists in business object meta-data and as such considered softly defined. This allows you to have completely different lifecycle rules for business objects belonging to the same maintenance object.

Here are examples of two business objects with different lifecycles thast belong to the same maintenance object.


Copyright © 2026, Oracle and/or its affiliates.




The boxes in the above diagram show the potential valid states a business entity of the above business object can be in. The lines between the boxes indicate the state transition rules. These rules govern the states it can move to while in a given state. For example, the above diagram indicates a high bill complaint that's in the **Lodged** state can be either **Canceled** or moved into the **Preliminary Investigation** state.

When you set up a business object, you define both its valid states and the state transition rules, including the label for the Action to display on a button for users that may initiate a transition. For example, the action label to transition to the 'Approved' state could be configured as 'Approve'.

> ⓘ **Note**
>
> The status description and the action label of base delivered values may be overridden by your implementation.

> ⓘ **Note**
>
> A Maintenance Object supports soft lifecycle rules if it is defined with a **Status Field** Maintenance Object option and that field is not a lookup field (which denotes a hard-coded lifecycle).

The topics that follow describe important lifecycle oriented concepts.

- One Initial State and Multiple Final States
- State-Specific Business Rules
- Inheriting Lifecycle
- Auto-Transition
- Keeping An Entity In Its Last Successful State
- Monitoring Batch Processes
- Transitory States


Copyright © 2026, Oracle and/or its affiliates.

- State Transitions Are Audited
- Required Elements Before Entering A State
- Capturing a Reason for Entering a State
- Opening UI Maps Before Entering A State

## One Initial State and Multiple Final States

When you set up lifecycle states, you must pick one as the initial state. The initial state is the state assigned to new entities associated with the business object. For example, the above high-bill complaint business object defines an initial state of **Lodged**, whereas the literature request one defines an initial state of **Literature Sent**.

You must also define which statuses are considered to be "final". Typically when an entity reaches a "final" state, its lifecycle is considered complete and no further processing is necessary.

> **Note**
>
> **Allowing An Entity To Be "Reopened".** You can set up your state transition rules to allow a business entity to be "reopened" (i.e., to be moved from a final state to a non-final state). Neither of the above examples allows this, but it is possible if you configure your business object accordingly.

## State-Specific Business Rules

For each state in a business object's lifecycle, you can define the following types of business rules.

| System Event | Optional / Required | Description |
|---|---|---|
| **Enter** | Optional | Algorithms of this type apply business rules when a business object instance enters a given state. |
| **Exit** | Optional | Algorithms of this type apply business rules when a business object instance exits a given state. |
| **Monitor** | Optional | Algorithms of this type monitor a business object instance while in a given state. Typically these are used to auto-transition it to another state. |

The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object Inheritance for more information.

> **NOT_SUPPORTED**
>
> Refer to BO Algorithm Execution Order for a summary of when these lifecycle algorithms are executed with respect to BO level algorithms.


Copyright © 2026, Oracle and/or its affiliates.

- Entering A State
- Exiting A State
- Monitor Rules

## Entering A State

You can define algorithms that execute before a business entity enters a given state. For example, you could develop an algorithm that requires a cancellation reason before an entity is allowed to enter the **Canceled** state.

You can also incorporate state auto-transitioning logic within this type of algorithms. Refer to auto-transition for more information.

Also note that when a record is processed by the monitor batch program, by default the BO Post Processing, BO Audit and MO Audit algorithms are not executed. However, it is possible for an enter algorithm to indicate that the other algorithms should be executed by the batch process by setting the “force post processing” indicator to true.

## Exiting A State

You can define algorithms that execute when a business entity exists a given state. For example, you could develop an algorithm that clears out error messages when a given entity exits the **Error** state.

Also note that when a record is processed by the monitor batch program, by default the BO Post Processing, BO Audit and MO Audit algorithms are not executed. However, it is possible for an exit algorithm to indicate that the other algorithms should be executed by the batch process by setting the “force post processing” indicator to true.

## Monitor Rules

You can define algorithms to monitor a business entity while it is in a given state. This type of logic is typically used to check if the conditions necessary to transition the entity to another state exist (and, if so, transition it). For example, transition an entity to the **Canceled** state if it's been in the **Error** state too long. Another common use is to perform ancillary work while an entity is in a given state. For example, update statistics held on the object while it's in the **Active** state.

Monitor algorithms are invoked when a business entity first enters a state and periodically after that in batch. You have the option to defer the monitoring of a specific state until a specific monitoring batch job runs. You do so by associating the state with a specific monitoring process. In this case the system will only execute the monitoring rules of this state when that specific batch process runs. This is useful when processing one type of record typically creates another type of record. You may want the processing of the second set of records to be deferred to a later time.

A monitor algorithm can carry out any business logic. In addition it can optionally tell the system to do either of the following:

- Stop monitoring and transition to another state. The system will not call any further monitoring algorithm plugged in on the state and attempt to transition the entity to the requested new state.
- Stop monitoring. Same as above except that no transition takes place. You may want to use this option to prevent transitions while some condition is true.

If none of the above is requested the system keeps executing subsequent monitoring algorithms.


Copyright © 2026, Oracle and/or its affiliates.

Also note that when a record is processed by the monitor batch program, by default the BO Post Processing, BO Audit and MO Audit algorithms are not executed. However, it is possible for a monitor algorithm to indicate that the other algorithms should be executed by the batch process by setting the “force post processing” indicator to true.

> **NOT_SUPPORTED**
>
> Refer to Business Object - Lifecycle for more information on how to set up state-specific algorithms.

## Inheriting Lifecycle

If a business object references a parent business object, it always inherits its lifecycle from the highest-level business object in the hierarchy. In other words, only the highest-level parent business object can define the lifecycle and the valid state transitions for each state. Child business objects, in all levels, may still extend the business rules for a given state by introducing their own state-specific algorithms.

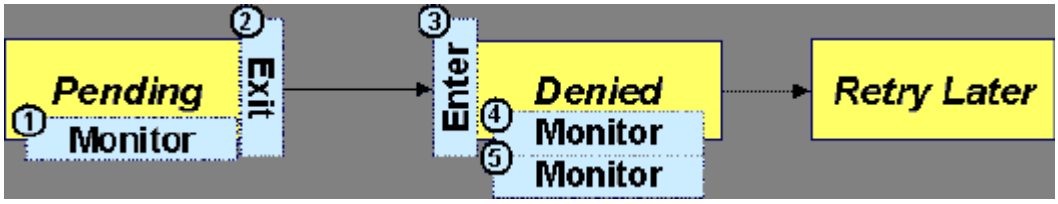
The system executes all state-specific algorithms at all levels in the inheritance chain starting from the highest-level parent business object moving on to lower levels.

## Auto-Transition

In a single transition from one state to another, the system first executes the **Exit** algorithms of the current state, transitions the entity to the new state, executes the **Enter** algorithms of the new state followed by its **Monitor** algorithms. At this point if a **Monitor** algorithm determines that the entity should be further automatically transitioned to another state the remaining monitoring algorithm defined for the current state are not executed and the system initiates yet another transition cycle.

Notice that an **Enter** algorithm can also tell the system to automatically transition the entity to another state. In this case the remaining **Enter** algorithm as well as all **Monitor** algorithms defined for the current state are not executed.

The following illustration provides an example of an auto-transition chain of events.




In this example a business entity is in a Pending state. While in that state a **Monitor** algorithm determines to auto-transition it to the Denied state. At this point the following takes place:

- No further **Monitor** algorithms of the Pending state are executed
- Pending state **Exit** algorithms are executed
- The system transitions the entity to the Denied state
- Denied state **Enter** algorithms are executed. No further auto-transition is requested.
- Denied state **Monitor** algorithms are executed. No further auto-transition is requested.


Copyright © 2026, Oracle and/or its affiliates.

## Keeping An Entity In Its Last Successful State

By default, any error encountered while transitioning a business entity from one state to another rolls back all changes leaving the entity in its original state.

When applicable, the Maintenance Object can be configured to always keep an entity in its last successful state rather than rolling all the way back to the original state. This practice is often referred to as "taking save-points". In case of an error, the entity is rolled back to the last successfully entered state and the error is logged on the maintenance object's log. Notice that with this approach no error is returned to the calling process, the error is just logged.

The logic to properly log the error is in a **Transition Error** Maintenance Object plug-in. The system considers a maintenance object to practice "save-points" when such an algorithm is plugged into it.

Even if the maintenance object practices "save-points", in case of an error the system will not keep an entity in the last successfully entered state if that state is either marked as transitory or one of its **Enter** algorithms has determined that the entity should proceed to a next state. The system will roll back to the first previous state that does not match these conditions.

## Monitoring Batch Processes

A monitor batch process may be used to transition a business object into its next state by executing the monitor algorithms associated with the current state of the entity. The use cases for performing the monitor logic in batch are as follows:

- The record may be waiting for something else to occur before transitioning. The monitor algorithm may be coded to determine if the condition is satisfied and initiate the transition then. For example perhaps when entering a state, a field activity is generated and the record should exit the state when the field activity is complete. The monitor algorithm can check the status of the field activity.
- Perhaps a record is added or updated manually and the next step in the BO lifecycle includes a large amount of processing such that the logic should occur in batch. In this case the BO status is configured with an explicit reference to a batch control (referred to as "deferred"), which indicates to the system that the monitor algorithms should not be performed automatically (but should be deferred to batch). Later when the batch process runs, it selects all the records to process to progress the records.

  > **Note**
  >
  > When a status includes a deferred batch control, it may also be configured to allow a user to manually transition the record to the next state rather than waiting for batch. When a user manually transitions a record that includes monitor algorithms, those algorithms are not executed.

- Perhaps a record is added or updated in batch, but a subsequent step in the overall lifecycle should be processed later. This may be accomplished by ensuring that the batch control linked to the state to process later does not match the batch control that added or updated the record.
- Monitor processes may also be used to periodically perform some logic related to the record without actually transitioning the record.

Note that only the parent business object may refer to a deferred monitor batch process. However, any business object in the "inheritance" chain may be configured with monitor algorithms, which will all be executed.


Copyright © 2026, Oracle and/or its affiliates.

The base package provides a periodic monitoring batch process for each maintenance object that supports a configurable BO lifecycle. The process periodically executes the monitoring algorithms associated with the current state of an entity, excluding states explicitly referencing a deferred monitoring batch process that is for a different batch control.

A deferred monitoring process selects records whose current state references this particular batch control as their monitor process. Deferred monitor process is only needed when an object has different use cases for monitoring the same type of records with different schedules. In this case only one periodic monitor batch should be configured. Other monitors should be configured to restrict by batch control so that there is no overlap in processing.

> **Note**
>
> **MO option configuration**. The maintenance object includes options to indicate the batch controls delivered for periodic and deferred monitor batch controls.

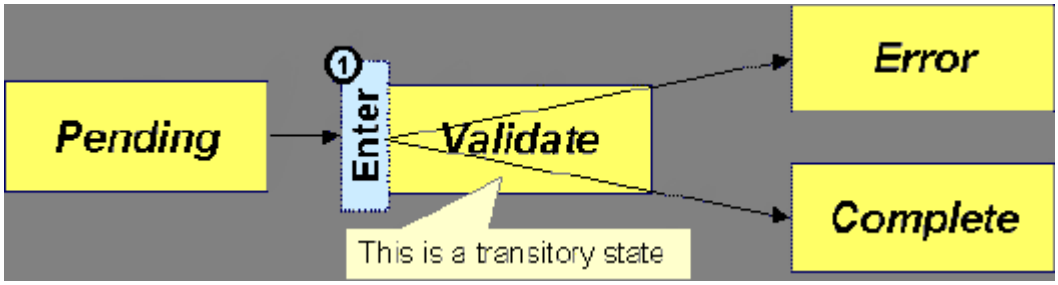
Your business rules will dictate the execution frequency of each monitoring process and the order in which they should be scheduled. Refer to Monitor Background Processes in the background process chapter for more information about the parameters supported for this type of batch process.

> **Note**
>
> **Updates to the business object.** When the monitor algorithms indicate that the business object should transition, the monitor batch processes are responsible for ensuring the business object is transitioned appropriately and that the appropriate exit, enter and monitor algorithms are executed. Please note that the business object is not updated using a call to the maintenance object service and therefore by default the business rules plugged in to the business object are not executed. However, it is possible for an Enter algorithm, Exit algorithm or Monitor algorithm to indicate that the other algorithms should be executed by the batch process. If the "force post processing" indicator is set to true, then the batch process invokes the BO Post Processing, BO Audit and MO Audit algorithms.

## Transitory States

You can define a state as **Transitory** if you do not wish the business entity to ever exist in that particular state.

The following illustrates a lifecycle with a transitory Validate state.




In this example, the business entity is saved still not validated in the Pending state. At some point, the user is ready to submit the entity for validation and transitions it into a transitory state Validate whose **Enter** rules contain the validation logic. The responsibility of the transitory


Copyright © 2026, Oracle and/or its affiliates.

state's **Enter** algorithms is to decide if the entity is valid or in error and then transitions it into the appropriate final state. In this scenario, you may not ever want the business entity to exist in the Validate state.

Let's also assume that the maintenance object in this example is practicing "save-points" and requires the entity to be kept in its last successful state. If an error were to occur during the transition from **Validate** to the next state, the system would roll back the entity back to Pending, and not Validate even though the entity has successfully entered the Validate state. Refer to the Auto Transition section for more information.

## State Transitions Are Audited

Most Maintenance Objects that support soft lifecycle definition also have a log to hold significant events throughout a business entity's lifecycle. For example, log entries are created to record:

- When the business entity is created (and who created it)
- When its status changes (and who changed it)
- If a transition error occurred (and the error message)
- References to other objects created throughout the entity's lifecycle. For example, if a To Do entry is created as part of processing the entity, the To Do Entry is referenced in the log.
- Manual entries added by a user (think of these as "diary" entries)

When a business entity is first created and when it transitions into a new state the system calls **Transition** algorithm(s) plugged in on the Maintenance Object to record these events. If the maintenance object supports a log these events can be captures as log entries.

> **Note**
>
> Most base package maintenance objects supporting a log may already provide state transition logging as part of their core logic. In this case you only need to provide a **Transition** plug-in if you wish to override base logging logic with your own.

## Required Elements Before Entering A State

You can define additional elements that are required before a business entity can enter a given state. For example, let's assume that a Cancel Reason must be defined before an object can enter the *Canceled* state. You do this by indicating that element as a **Required Element** state-specific option on the appropriate state on the business object.

## Capturing a Reason for Entering a State

Some business objects support configuring certain states to allow or require a status reason when an object enters the state. The product provides a centralized status reason table that may be used to define the valid BO status reasons for various business objects and various states. The status reasons are defined using the Status Reason portal.

The following sections provide additional information about the BO status reason functionality.


Copyright © 2026, Oracle and/or its affiliates.

### Maintenance Object Must Support Status Reason

In order for a business object to use the centralized status reason table to define reasons, the maintenance object must first support the status reason. MOs that support status reason have the following characteristics:

- The primary table includes a column for Status Reason. This represents the status reason for the record's current status, if applicable.
- The log table includes a column for Status Reason. The standard logic for capturing a log record when entering a state also captures the status reason, if applicable. This allows a user to review the history of the changes in status and the status reason captured for a previous state transition, if applicable.
- The maintenance object option collection includes an option that defines the Status Reason field. This setting is a trigger for business objects of this MO to be able to configure states to allow or require status reason.

### Business Object State Indicates if Reasons are Applicable

Once the MO is configured to support status reason, configuration on the business object is required to indicate the states where a reason is applicable. States may be configured to require status reasons, allow status reasons as optional or not allow status reasons. With this configuration, the framework will automatically get the list of valid reasons for a state that allows or requires them and then prompt the user for a status reason when a manual state transition occurs for that state. It also automatically triggers an error if the state requires a status reason and no reason is provided.

> **Note**
>
> The status reason configuration on the business object state is customizable. That means that for a product owned business object, an implementation may opt to change the delivered configuration.

Status reasons are defined for the parent (or "lifecycle") business object. All business objects in the hierarchy of the parent business object have the same valid reasons for their states.

The status reason code must be unique for the centralized status reason table. Business object and status are required fields, so it is not possible to share a common reason code (like "Not applicable") across multiple business objects or states. If multiple BOs / states want to support a reason "Not applicable" then each must define a unique record for it. This point should be considered when planning for your status reasons.

### Override Description

The description of product owned status reasons may be overridden by your implementation.

### Selectable vs. Not Selectable

When defining a status reason, you may indicate whether it's **Selectable** or **Not Selectable**. When a manual transition is performed and a user is prompted for a status reason, only the **Selectable** reasons are presented. The **Not Selectable** reasons may be defined to support transitions that occur via algorithm processing.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The Selectable setting is customizable. That means that if a product provides a base owned status reason for a business object state, an implementation may opt to change whether it is selectable or not. Careful consideration should be made before changing a base delivered status reason from **Not Selectable** to **Selectable** as this may affect base provided algorithm functionality that could be relying on the setting of **Not Selectable**.

#### Status Reason Business Object

The status reason maintenance object, as with many maintenance objects in the product, references a business object used to define attributes and behavior related to defining status reasons. The framework provides a business object for status reason (`F1-BOStatusReason`). For the business objects that have states that require a status reason (let's call these "transactional BOs"), if there is some special logic required for defining the status reasons, it is possible to define a different status reason BO. In this situation, the override status reason BO to use for capturing status reasons should be defined as a BO option on the transactional BO using the **Status Reason Business Object** option type. If a transactional BO does not define any status reason BO option, then the `F1-BOStatusReason` is used when adding a status reason.

#### Defining a Usage

The base product status reason BO provides the ability to define a "usage" value. This is useful for algorithms that perform state transitions where a status reason is needed and where the algorithm is usable by more than one business object. In this case, the status reason to use cannot be provided as a parameter because each business object must define its own set of status reasons for each state. The Usage value can be used instead. Each business object can configure the status reason to use in the algorithm and set the appropriate usage value. The algorithm can reference the usage value and retrieve the correct status reason to use based on the record's transactional BO.

The status reason business object provided with the framework product (`F1-BOStatusReason`) supports capturing a usage. The valid usage values are defined in the **Status Reason Usage** characteristic type.

#### Alternatives for Defining Reasons

There may be business objects in the system that capture reasons that are defined somewhere besides the BO status reason table. For example, some objects may have an explicit administrative table for status reasons. Some objects may use a Lookup or an Extendable Lookup to capture reasons. Refer to the business object description for information about how valid reasons are defined, if applicable.

If a business object supports a reason that is not related to a state transition (such as a creation reason), the BO status reason would not be used. One of the alternate methods for defining a reason, described above, would be used.

## Opening UI Maps Before Entering A State

You can define a UI Map to capture additional elements before a business entity can enter a given state. You do this by configuring the **State Transition UI Map** state-specific option on the appropriate state on the business object.


Copyright © 2026, Oracle and/or its affiliates.

There may be circumstances in which the status pre-processing logic can determine the value of the elements, in which case it is not necessary to invoke the map. The system provides a data area (`F1-StateTransitionCommon`) which contains a boolean element to indicate if the state transition map can be skipped. This data area can be included in your pre-processing script to allow the element to be set according to your business logic. The base maintenance script (`F1-MainProc`) references this element to determine whether to open the state transition map.

## BO Algorithm Execution Summary

This table highlights the processing steps that occur when adding or changing a record that is governed by a business object.

| Invoke BO | |
|---|---|
| **Event** | **Comments** |
| BO Pre-processing algorithms executed | These algorithms are only executed when Invoke BO is used. The business object in the Invoke BO is the one whose rules are executed. |

| MO Processing | |
|---|---|
| **Event** | **Comments** |
| Determine if status has changed. | The system keeps a note of the new status value but initially proceeds with the old value. |
| MO Processing. | Standard MO processing, including MO validation is executed. |
| Determine BO algorithm executed. | The MO level algorithm is executed to determine the identifying BO. |
| BO Validation algorithms executed. | |
| State transition rules are performed if the status has changed. | BO Status Exit algorithms for the "old" status executed.<br>Status updated to the new value.<br>BO Status Enter algorithms for the "new" status executed.<br>If no error â€” MO Transition algorithms are executed. |

**NOT_SUPPORTED**

Refer to State Transitions are Audited for more information.


Copyright © 2026, Oracle and/or its affiliates.

| MO Processing | |
|---|---|
| | If error and there are "save points" the MO Transition Error algorithms are executed.<br><br>**NOT_SUPPORTED**<br>Refer to Keeping An Entity In Its Last Successful State for more information.<br><br>Otherwise, the error is reported. |
| BO Status Monitor algorithms are executed. | If the record transitions again, the prior step (State transition rule step) is repeated for the new transition. |
| BO Post-processing algorithms are executed. | |
| BO Audit algorithms are executed. | These algorithms are only executed if the system detects a change in elements that are not marked with "no audit". |

**Note**

To emphasize, the steps in the MO Processing table are only executed when the maintenance object service is invoked. Any add or update initiated by an "invoke BO" statement will invoke the MO service. This is also true for web service that invoke the business object. The Monitor Batch Process does not invoke the maintenance service. By default the monitor batch process only executes the monitor algorithms and the state transition rules (if the monitor algorithms indicate that a status change should occur). However, it is possible for an Enter algorithm, Exit algorithm or Monitor algorithm to indicate that the other algorithms should be executed by the batch process. If the "force post processing" indicator is set to true, then the batch process invokes the BO Post Processing, BO Audit and MO Audit algorithms.

**Note**

For records that do not have a status, the state transition rules and the monitor rules are not applicable.

## Granting Access To Business Objects

Every business object must reference an application service. When you link a business object to an application service, you are granting all users in the group access to its instances. You can prevent users from transitioning a business object into specific states by correlating each business object status with each application service action (and then don't give the user group rights to the related action).


Copyright © 2026, Oracle and/or its affiliates.

**NOT_SUPPORTED**

Refer to The Big Picture Of Application Security for information about granting users access rights to an application service.

The system checks if a user has access rights each time the application is invoked to add, change, delete, read, or transition a business object. However, if a business object invokes another business object, we assume that access was controlled by the initial business object invocation and we do not check access for other business objects that it invokes. In other words, access rights are only checked for the initial business object invoked in a service call.

In order to apply business object security the system must be able to determine the business object associated with the actual object being processed. To do that the Maintenance Object itself has to have a **Determine BO** algorithm plugged in. If this algorithm is not plugged in or it cannot determine the BO on the MO, the system will not invoke any BO rules. If the business object cannot be determined for a maintenance object instance, business object security is not checked. In this case the system checks the user's access rights using standard maintenance object security.

**Note**

**Parent business objects are ignored.** If a child business object exists, a user need only have access to the child business object's application service (not to every application service in the business object hierarchy).

# Defining Business Objects

This portal is used to view and maintain information about a business object.

You can access the portal from the **Admin Menu**, and then **System**, and then **Business Object**. You are brought to a query portal with options for searching for a specific business obejct. Once a business object has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding Business Objects for more information.

**Caution**

Important! If you introduce a new business object, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Business Object.** This zone provides information about the selected business object.
- **Schema Editor.** This zone describes the business object’s schema and supports basic edits in plain text. Refer to the **Schema** tab below for additional schema editor information.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the business object's schema using the Schema Designer tool. The purpose of the schema is to describe the business object's properties and map them to corresponding maintenance object fields.


Copyright © 2026, Oracle and/or its affiliates.

> **NOT_SUPPORTED**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

> **Tip**
>
> Click the **View Schema** hyperlink to view the business object's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **Algorithms** tab page. Refer to A Business Object May Define Business Rules for more information.

- **Algorithms.** This zone is used to maintain the algorithms associated with the current business object.
- **Hierarchy Algorithms.** This zone displays all algorithms associated with the business object's inheritance hierarchy. The zone is visible when the current business object is part of a hierarchy and such algorithms exist.

The following zones may appear as part of the portal's **Options** tab page. Refer to A Business Object Defines Options for more information.

- **Options.** This zone is used to maintain various configuration options associated with the current business object.
- **Hierarchy Options.** This zone displays all options associated with the business object's inheritance hierarchy. The zone is visible when the current business object is part of a hierarchy and such options exist.

The following zones may appear as part of the portal's **Lifecycle** tab page. This tab is only available when the maintenance object supports lifecycle definition.

- **Lifecycle.** This zone describes and maintains the current business object's lifecycle definition. Broadcast a specific status to review its complete configuration.
- **Status Details.** This zone is used to maintain basic definition information for a selected status within the current business object's lifecycle.
- **Status Algorithms.** This zone is used to maintain algorithms for a selected status within the current business object's lifecycle.
- **Status Options.** This zone is used to maintain various configuration options for a selected status within the current business object's lifecycle.
- **Hierarchy Status Algorithms.** This zone displays all algorithms associated with the business object's inheritance hierarchy for a selected status. The zone is visible when the current business object is part of a hierarchy and such algorithms exist.
- **Hierarchy Status Options.** This zone displays all options associated with the business object's inheritance hierarchy for a selected status. The zone is visible when the current business object is part of a hierarchy and such options exist.
- **Status Reasons.** This zone is used to maintain reasons for a selected status. The zone is visible when the maintenance object supports a reason field and usage of reasons is allowed by the status definition.

The following zones may appear as part of the portal's **Lifecycle View** tab page. This tab is only available when the maintenance object supports lifecycle definition.


Copyright © 2026, Oracle and/or its affiliates.

- **Lifecycle View.** This zone provides a schematic view of the current business object's lifecycle definition. Broadcast a specific status to review its complete configuration.
- **Status Details.** This zone is used to maintain basic definition information for a selected status within the current business object's lifecycle.
- **Status Algorithms.** This zone is used to maintain algorithms for a selected status within the current business object's lifecycle.
- **Status Options.** This zone is used to maintain various configuration options for a selected status within the current business object's lifecycle.

The following zones may appear as part of the portal's **Hierarchy** tab page. This tab is only available when the current business object is part of a hierarchy.

- **Business Object Hierarchy.** This zone lists the inheritance hierarchy of business objects up to and including the current business objects. The zone is visible if the current business object is not the top level business object of the hierarchy.
- **Business Object Tree.** This zone describes the entire hierarchy of the current business object.
- **Hierarchy Algorithms.** This zone lists all algorithms defined for the current business object. The zone is visible if algorithm information exists for the current business object.
- **Hierarchy Lifecycle Algorithms.** This zone lists all algorithms associated with the current business object's inheritance hierarchy. The zone is visible if the current business object is not the top level business object of the hierarchy and algorithm information exits. Refer to Inheriting Lifecycle for more information.
- **Hierarchy Options.** This zone lists all configuration options for the current business object's lifecycle. The zone is visible if options information exists for the current business object.
- **Hierarchy Lifecycle Options.** This zone displays all options associated with the current business object's inheritance hierarchy. The zone is visible if the current business object is not the top level business object of the hierarchy and options information exits. Refer to Inheriting Lifecycle for more information.

The following zones may appear as part of the portal's **References** tab page.

- **Business Object References.** This zone list various configuration entities referencing the current business object or its schema.

### Actions

The page supports the following special actions:

- **Generate User Interfact Configuration Artifacts.**When creating a new buisiness object or editing an existing one, you may request to generate the user interface configuration artifacts needed to properly maintain and display the business object on an existing portal. For example, if a display script is needed to further prepare the business object's details for presentation then request to generate such script. You can then navigate to the script from the **User Interface Configuration** section and adjust it as needed.
- **Subclass a Business Object.** Use the **Subclass** action to create a new business object that is a child of an existing business object. The action generates the new business object's schema as inherited from the existing business object. Refer to Business Object Inheritance for more information.
  Once created, you may extend and adjust the new business object's schema and rules as needed.


Copyright © 2026, Oracle and/or its affiliates.

- **Create an Algorithm.** Use this action, on the Algorithm tab or the Lifecycle tab for a specific status, to creates a script based algorithm related to this business object. You are prompted for information regarding the plug-in spot (BO or BO Status) and the system event, the name, description, etc. Once all the information is provided, the system creates an algorithm type, a script, an algorithm and links the algorithm to the business object. You may then navigate to the script to adjust it as needed.
- **Inactivate / Activate Base Algorithms.** You can inactivate algorithms on base Business Objects. Your implementation may want to use a business object provided by the base product, but may want to inactivate one or more algorithms provided by the base business object. To do that, use the "Inactivate" action on the appropriate algorithm entry on the Algorithm tab. Use use the "Activate" action to mark an inactive algorithm as active. Note that these actions add and remove the "Inactive Algorithm" businesss option as needed.
- **View Schema.** Click the hyperlink to view the business object's expanded schema definition. Doing this opens the schema viewer window.

# Advanced BO Tips and Techniques

The topics in this section describe some advanced tips and techniques for configuring business objects.

- Creating a Physical Business Object
- Managing To Do Entries
- Submitting a Batch Job

## Creating a Physical Business Object

The physical business object's schema represents the complete physical structure of the maintenance object, and includes elements for all fields in the maintenance object's tables. There are several processes in the system that may use a maintenance object's physical BO.

To create a physical business object for the maintenance object:

1. Navigate to Business Objectand add a business object for the maintenance object. Do not fill in any of the UI oriented configuration.
2. The business object is created with a schema that looks like the physical structure of the maintenance object.
3. Save the physical business object.
4. Navigate to Maintenance Object and search for the maintenance object.
5. On the **Option** tab, add a new option with the type **Physical Business Object**. The value is the name of the physical business you just created.

## Managing To Do Entries

The product provides several base algorithm types that may be used to manage To Do entries through status changes for a given record via BO lifecycle plug-ins.

### Create To Do Entry

The product supplies a BO status Enter algorithm type **Generic To Do Creation** (`F1-TDCREATE`) that creates a To Do entry based on parameter configuration. Refer to the algorithm type description for more information about how it determines the To Do type or To


Copyright © 2026, Oracle and/or its affiliates.

Do role and how to populate the appropriate message text onto the To Do. This algorithm may be used in conjunction with the Retry Logic (below).

If your implementation has a business rule that requires a To Do entry to be created when entering a given BO status and the logic provided by the algorithm type meets the needs of the business rule, this algorithm type may be used. Create an algorithm for the algorithm type, populate the algorithm parameters according to the business rules and plug the new algorithm into the appropriate business object status as an Enter algorithm.

**Retry Logic**

The algorithm type **Retry for To Dos** (`F1-TODORETRY`) is supplied for a special use case. It is a BO status monitor plug-in and may be used for a state that is a type of 'error' or 'waiting' state. It relies on the To Do entry creation logic to set a Retry Frequency. The algorithm transitions to the originating state to retry the logic. The idea is that the condition that caused the record to enter the 'error' or 'waiting' state may be resolved after some period of time has passed, allowing the record to progress in its lifecycle. Refer to the algorithm type description for more information about its logic.

To use this functionality, create an algorithm for this algorithm type, populate the algorithm parameters according to the business rules and plug the new algorithm into the appropriate business object status as a Monitor algorithm. The state should also have an algorithm for the **Generic To Do Creation** algorithm type plugged in as an Enter algorithm (or something equivalent) that sets the appropriate Retry Frequency.

**To Do Completion**

It is common that one or more To Do entries associate with a given record should be completed when exiting a state (if it is not already completed). The system supplies the algorithm type **Generic To Do Completion** (`F1-TODOCOMPL`) that may be used for this purpose. Note that the algorithm type functionality is not tied to any To Do creation logic. It may be used for any use case where To Do entries should be completed on exiting a state. Refer to the algorithm type description for more information about its functionality and how to prevent certain To Do entries from being automatically completed.

To use this functionality, create an algorithm for this algorithm type, populate the algorithm parameters according to the business rules and plug the new algorithm into the appropriate business object status as an Exit algorithm.

## Submitting a Batch Job

The product provides a base algorithm type that submits a batch job when entering a BO state. This functionality allows for "event driven" batch submission where the event is the lifecycle transition for a certain record.

The algorithm type is **Create Batch Job Submission Entry for Batch Control** (`F1-SCHEDJOB`). The batch control code is a parameter for the algorithm. Refer to the algorithm type description for more information about its logic.

To use this functionality, create an algorithm for this algorithm type, populate the algorithm parameter with the batch control that should be submitted and plug the new algorithm into the appropriate business object status as an Enter algorithm.

# Defining Status Reasons

Status Reasons are used to provide more information about why a business object transitioned to a given state. The status reason table provides a centralized place where status reasons can be defined across many different business objects and states.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to Defining Reasons for Entering a State for overview information.

If a business object has one or more states that are configured to capture a status reason, you may configure the valid reasons by navigating to the status reason portal using **Admin**, and then **System**, and then **Status Reason**.

The topics in this section describe the base-package zones that appear on the Status Reason portal.

#### Business Objects with Status Reason List

This zone displays the business objects that have one or more status values that allow status reasons to be defined.

Click the broadcast icon to open other zones that contain more information about the business object's status reasons.

#### Status Reasons

The **Status Reasons** zone contains a list of the existing status reasons for the broadcasted business object.

# Data Areas

The data area has no business purpose other than to provide a common schema location for re-used schema structures. It exists solely to help eliminate redundant element declaration. For example, if you have multiple schemas that share a common structure, you can set up a stand-alone data area schema for the common elements and then include it in each of the other schemas.

Be aware that a stand-alone data area can hold elements that are mapped to true fields. For example, you might have 50 different types of field activities and all might share a common set of elements to identify where and when the activity should take place. It would be wise to declare the elements that are common for all in a stand-alone data area and then include it in the 50 field activity business objects.

It's strongly recommended that you take advantage of stand-alone data areas to avoid redundant data definition!

> **Caution**
>
> **Dynamic inclusion!** When the system renders a schema, all schemas included within it are expanded real-time. This means that any change you make to a data area will take effect immediately within all schemas it is referenced within. Therefore, changing a schema that is referenced in a large number of schemas may be too large to be performed online. In this situation, the change should be performed in batch. Refer to "Update Object Schema (F1-SCMUP)" batch control for more information.

This section describes additional topics related to designing and working with data areas.


Copyright © 2026, Oracle and/or its affiliates.

### Dynamic Extension

You may use a custom data to extend a base product data area. To extend another data area, reference that data area in the **Extended Data Area** field. By extending a data area you can add additional elements to a base product data area.

Here's an example of an extended data area:

- The product releases with data area A, which contains elements a, b, and c.
- Your implementation creates data area CM-A, which contains element z, and references data area A as the extended data area.
- At run time, everywhere data area A is included it will contain elements a, b, c, and z.
- Defining Data Areas

# Defining Data Areas

Use this page to define basic information about a data area.

You can access the portal from the **Admin Menu**, and then **System**, and then **Data Area**. You are brought to a query portal with options for searching for a specific data area. Once a data area has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a new data area, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Data Area.** This zone provides information about the selected data area.
- **Extending Data Areas.** This zone lists all data areas extending the current data area. The zone appears if such data areas exist.
- **Schema Editor.** This zone describes the data area's schema and supports basic edits in plain text. Refer to the **Schema** tab below for additional schema editor information.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the data area's schema using the Schema Designer tool. The purpose of the schema is to describe the data area's properties and map them to the underlying internal service fields .

> **NOT_SUPPORTED**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.


Copyright © 2026, Oracle and/or its affiliates.

> **Tip**
>
> Click the **View Schema** hyperlink to view the data area's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **References** tab page.

- **Data Area References.** This zone list various configuration entities referencing the current data area or its schema.

#### Actions

The page supports the following special actions:

- **View Schema.** Click the hyperlink to view the data areas's expanded schema definition. Doing this opens the schema viewer window.

# Business Services

A business service is used to expose a back-end service so that it may be invoked by a script or a zone or a map to retrieve information or perform functions, depending on the related service.

As with the business object, the business service's interface to the internal service is defined using its schema. The schema maps the business service's elements to the corresponding elements in the internal service program's XML document. Just as a business object can simplify the schema of its maintenance object by only defining elements that it needs and "flattening" entries in a child collection to be defined as a singular element, a business service schema may simplify its service XML in a similar way.

> **NOT_SUPPORTED**
>
> Refer to Schema Syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

Inbound web services and scripts support interaction with business services. You can also invoke a business service from a Java class.

- Defining Service Programs
- Defining Business Services

## Defining Service Programs

This transaction defines services available in the system. These include user interface services as well as stand-alone services that perform a specific function. A service may be referenced by a business service. Use this transaction to view existing service and introduce a new stand-alone service to be made available to a Business Service.

You can access the portal from the **Admin Menu**, and then **System**, and then **Service Program**. You are brought to a query portal with options for searching for a specific service program. Once a service program has been selected you are brought to the maintenance portal to view and maintain the selected record.


Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! When adding new service programs, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Service Program.** This zone provides information about the selected service program.
- **Service Program References.** This zone list various configuration entities referencing the current service program.

**Application Service** is the application service that is used to provide security for the service. If the service is related to a maintenance object, the access modes for the application service should be the standard **Add**, **Change**, **Delete** and **Inquire**. For other services, the application service should have the **Execute** access mode.

**Service Type** indicates whether the service is a **Java Based Service** or a **Java (Converted) Service**. Note that services generated to support a portal in the system will not have a service type populated.

The **Programs** grid shows the list of program user interface components associated with the service. For a stand-alone service, this list is typically not applicable.

# Defining Business Services

Use this page to define basic information about a Business Service.

You can access the portal from the **Admin Menu**, then **System**, then **Business Service**. You are brought to a query portal with options for searching for a specific business service. Once a business service has been selected you are brought to the maintenance portal to view and maintain the selected record.

> **⚠ Caution**
>
> Important! If you introduce a new business service, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Business Service.** This zone provides information about the selected business service.
- **Schema Editor.** This zone describes the business service's schema and supports basic edits in plain text. Refer to the **Schema** tab below for additional schema editor information.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the business service's schema using the Schema Designer tool. The purpose of the schema is to describe the business service's properties and map them to the underlying internal service fields.

> **ⓘ Note**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.


Copyright © 2026, Oracle and/or its affiliates.

> **Tip**
>
> Click the **View Schema** hyperlink to view the business service's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **References** tab page.

- **Business Service References.** This zone list various configuration entities referencing the current business service or its schema.

### Actions

The page supports the following special actions:

- **View Schema.** Click the hyperlink to view the business service's expanded schema definition. Doing this opens the schema viewer window.
- **Test.** Use this action to test your business service. Enter input values and save. The service is performed and output information is presented, but the system issues a roll-back so that the changes are not committed. Note that if your service uses the save-point dispatcher to perform any actions or if it issues a call to an external system, the roll-back does not apply to those actions. If the service issues an error the corresponding message is displayed. Note that the input and output testing user interface is dynamically rendered based on UIHints embedded in the business service's schema. As such, the user interface may be as elaborate or basic as the UIHints in that schema.

> **Note**
>
> This action is secured by a designated "Test a Service" application service and the "Generic Service Executor" application service and also requires security rights for creating a script, which is the alternative way for testing any service. It is recommended to allow access to this function in non-production environments only.

- Useful Services and Business Services

## Useful Services and Business Services

The following section highlights some business services and services provided by the product that may be useful for implementations to use.

### Data Explorer Service

The system provides a mechanism for performing an SQL select statement for use in scripting, Java plug-ins, or via a web service call. This is done by creating a zone using one of the data explorer zone types where the SQL is defined. Then, create a business service using the Data Explorer service (**FWLZDEXP**).

> **Note**
>
> There are numerous business services delivered with the base product that reference this service that may be used as a template.


Copyright © 2026, Oracle and/or its affiliates.

The following points highlight how to create your own business service for this service. Note that typically a separate business service exists for each zone.

- Enter a **Business Service** code and a **Description**. It is recommended to define the business service code to match the zone code so that it's easier to manage which business service invokes which zone.
- Select the **Service Name FWLZDEXP**.
- On the **Schema** tab, under the ***<schema>*** node, enter mapping for the fields that are required for the Data Explorer service:
  - The **Zone** should be mapped into service field ***ZONE_CD*** . Define the zone code as the default value.
  - For every **user filter** defined on the zone, create a schema mapping into the service field ***Fx_VALUE***, where "x" is the filter number (from the zone parameters).
  - For every **hidden filter** defined on the zone, create a mapping into the service field ***Hx_VALUE***, where "x" is the filter number (from the zone parameters).
  - The search results are returned as a list by the data explorer service. Each column value is in the service field ***COL_VALUE*** with an appropriate sequence number (***SEQNO***). The results can be flattened based on sequence number allowing for a logical element name to be defined.
  - Another useful field is ***ROW_CNT***, which provides the number of rows retrieved by your search.

The following is an example of the schema for a BS that receives a business object code and returns a list of status values and their descriptions that allow status reasons to be defined.

```
<schema>
     <zone mapField="ZONE_CD" default="F1-BOSTSLST"/>
     <bo mapField="H1_VALUE" />>
     <rowCount mapField="ROW_CNT"/>>
     <results type="list" mapList="DE">>
          <status dataType="string" mapField="COL_VALUE">
               <row mapList="DE_VAL">>
                    <SEQNO is="1" />>
               </row>>
          </status>>
          <description dataType="string" mapField="COL_VALUE">
               <row mapList="DE_VAL">>
                    <SEQNO is="2" />>
               </row>>
          </description>>
      </results>>
</schema>
```

### Maintenance Object Log Service

Many maintenance objects support a log table that follows a pattern of column names and behavior. The system provides a service called Generic MO Log Service (F1MOLOGP) that may be used to perform common functions related to log entries:

- Read log entries. If you pass a certain MO, primary key and log sequence number, the service will return the details of that log entry. The product provides a generic business service that may be used for this purpose — Generic MO - Retrieve Log Details (F1–


Copyright © 2026, Oracle and/or its affiliates.

ReadMOLog). Alternatively, it is possible to create a business service for a given MO where the MO code is assigned to the MO element using the default syntax. This allows business functionality specific to that maintenance object to use the specific BS.

- Add log. The service may be used to add a log entry. If a user log is added, then the comments from the user are populated in the detailed description. System generated log entries typically supply a message category / message number along with other information such as the status, a specific log type and optionally a related object reference (via a characteristic). The product provides a generic business service that may be used for this purpose — Add Generic MO Log (F1–AddMOLog). Alternatively, it is possible to create a business service for a given MO where the MO code is assigned to the MO element using the default syntax. This allows business functionality specific to that maintenance object to use the specific BS.

### Base Business Services

The following table highlights some business services provided by the product that may be useful for custom logic for an implementation.

> ⚠ **Caution**
>
> This is not intended to be a complete reference of Business Services. Refer to the business service page to find all the supported business services.

### Business Object Related Services

| Business Service Name | Description |
|---|---|
| F1-AutoTransitionBO | Performs monitoring algorithms associated with the current state of a given business object instance (which may result in subsequent state transitioning). |
| F1-CompareBusinessObjectData | Compares two versions of a given business object instance. |
| F1-DetermineBo | Determines the business object of a given instance of a maintenance object by executing the MO's Determine BO logic. |
| F1-GetRequiredFieldsForBOState | Returns the required fields for a given business object status. |
| F1-RetrieveBOOption | Returns BO option values for a given BO and option type. |
| F1-RetrieveBOStatusOption | Returns BO option values for a given BO, status and option type. |
| F1-RetrieveBOStatusOption | Retrieves a list of BOs for a given MO that are accessible for the current user. |
| F1-RetrieveBoStatusDescription | Return the description of a given BO status. |
| F1-RetrieveBusinessObjectLabel | Return the label appropriate for a given path (e.g. element) within a BO schema. |
| F1-RetrieveNextStates | Return a list of next possible states based on the input of a MO and its prime key, or a BO and one of its statuses. |

### Email Related Services


Copyright © 2026, Oracle and/or its affiliates.

| Business Service Name | Description |
|---|---|
| F1-EmailService | Sends an email message in real time. |
| F1-RetrieveEmailAddress | Retrieves the email addresses of users belonging to a To Do Role. |
| F1-RetrieveEnvironmentURL | Retrieves the current environment URL information for the installation. |

### Tools for Maps and Scripting

| Business Service Name | Description |
|---|---|
| F1-AddWarning | This business service may be used to log a warning by providing the message details. The system gathers all the warning and displays them to the user all at once. |
| F1-CheckOnlineConnection | Check if Connection is Online or Batch. |
| F1-CheckWarning | Used to check if any warnings have been accumulated. If warnings have been accumulated, any logic that cannot be rolled back (when the warnings are shown) should not be performed. |
| F1-DateMath | Performs various date and time math calculations. Refer to the BS description for more details. |
| F1-DateTimeFormattingService | Formats a given date / time based on the user's display profile settings. |
| F1-ExecuteScriptInNewSession | Executes a Service Script in a new processing session/transaction. |
| F1-GetFieldLabel | Retrieves the label for a given field. |
| F1-GetForeignKeyReference | Returns foreign key reference information for a given FK Reference and primary key, including info description, navigation option, and context menu. |
| F1-GetFKReferenceDetails | Returns foreign key reference information for a given MO and primary key, including FK reference code, info description, navigation option, search zone and context menu. |
| F1-GetLookupDescription | Returns lookup description for a lookup field value given the lookup field name. |
| F1-GetExtLookUpVal | Returns the list of values for a given extendable lookup BO. |
| F1-GetMonthInYearAbbreviation | Returns a 3-character month abbreviation for an input date in system format. |
| F1-NumberAmountFormatter | Formats a given amount or number based on the user's display profile settings. It also may receive input to adjust the scale and optionally apply currency settings. |
| F1-OutmsgDispatcher | Dispatches a real-time message giving the user the option of whether to persist the message on the database, and whether to trap errors that may take place during the call. Refer to Real Time Messages for more information. |


Copyright © 2026, Oracle and/or its affiliates.

| Business Service Name | Description |
|---|---|
| F1-OutmsgMediator | Alternative to F1-OutmsgDispatcher and may be a better option if the sender does not require an outbound message record to be instantiated. Refer to Real Time Messages for more information. |
| F1-RethrowError | Issues an application error using the input message category / number / parameters. |
| F1-RetrieveMODescription | Retrieves the description for a maintenance object. |
| F1-ReturnMessage | Returns the expanded message given a message category, number, parameters, and parameter types. |
| F1-SavePointDispatcher | Allows for a service script to be executed where exceptions are trapped and the transaction is rolled back to a save point set before the service script execution. |

**User Related Services**

| Business Service Name | Comments |
|---|---|
| F1-CheckApplicationSecurity | Checks a user's security for a given application service / access mode |
| F1-CheckUserAuthorization | Determine whether a given user is authorized for access based on the input application service, security code, and authorization level. |
| F1-DetermineIfUserCanApproveTD | Determine if the current user can approve a given To Do. |

# User Interface (UI) Options

Your implementation may create input forms and output maps that closely match your customer's business practice within the following user interface areas:

- Content presented in portal zones. For example, a map that displays the details of a business object or any other informational content. Refer to Map Zones for more information.
- An input form presented via Business Process Assistant (BPA) scripts for data entry and update purposes. For example, a form that captures business object information or prompts the user for additional information related to a task they perform.
- An input form presented as a panel within a process flow.

The HTML for these forms and maps may be provided using the following options:

- **UI Map.** Compose your HTML and store it as a UI map, which is a repository for an HTML document. The UI Map HTML gives you the ability to craft the display by any method that an HTML document can support, including JavaScript and full CSS functionality.
- **UI Hints.** With this technique you do not compose the HTML. Instead, the HTML is generated at run time based on a schema that describes the data being displayed and embeds user interface rendering instructions known as "UI Hints".
- Understanding UI Maps


Copyright © 2026, Oracle and/or its affiliates.

- Understanding UI Hints
- UI Hint Syntax
- Defining UI Maps

# Understanding UI Maps

The preferred method for creating input and output HTML maps is by leveraging the schema-based HTML rendering technique known as UI Hints. For more complex requirements, you may craft an HTML that closely matches your customer's business practices and capture it as a User Interface (UI) map.

The User Interface (UI) map is a repository for a single HTML document paired with an XML schema where the schema defines the data that the HTML document displays and/or modifies. The UI Map HTML gives you the ability to craft the display by any method that an html document can support, including JavaScript and full CSS functionality.

The topics in this section describe concepts and guidance relevant to UI map configuration.

### Output Maps and Input Forms

UI maps are typically crafted as output tables when used in conjunction with portal zones - please refer to Map Zones for more information. When referenced within BPA scripts or process flows, UI maps are typically crafted as forms for the capture and update of data.

**Figure HTML Input Form Rendered (for BPA Script)**

Copyright © 2026, Oracle and/or its affiliates.

**Figure HTML to Display Customer Business Object**

```
<html>
<head>
<title>Output Personal Information</title>
<link rel="stylesheet" type="text/css" href="cm_templates/cmStyles.css"/>
</head>
<body>

<table width="100%" border="0" cellpadding="2" style="margin-top:15px;" >

    <tr>
        <td/>
         <td/> <!-- locate the edit button on the bottom of the third column -->
        <td rowspan="99" style="vertical-align:bottom; margin-left:5px">
            <input type="button" value="edit" onClick="oraRunScript('HumanInfoU','personId');"/>
        </td>
    </tr>

    <tr>
        <td class="outputLabel">Name:</td>
        <td><span oraField="name" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Home Phone:</td>
        <td><span oraField="homePhone" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Business Phone:</td>
        <td><span oraField="businessPhone" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Cell Phone:</td>
        <td><span oraField="cellPhone" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">FAX:</td>
        <td><span oraField="fax" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Social Security:</td>
        <td><span oraField="socialSecurity" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Drivers License:</td>
        <td><span oraField="driversLicense" class="outputData"></span></td>
    </tr>
    <tr>
        <td class="outputLabel">Email:</td>
        <td><span oraField="email" class="outputData"></span></td>
    </tr>

</table>

</body>

<xml>
<root>
    <name>Greer,Johan</name>
    <email>jurgen.greer@media.com</email>
    <socialSecurity>939-30-3939</socialSecurity>
    <driversLicense>C8392020</driversLicense>
    <homePhone>(838) 030-0303</homePhone>
    <cellPhone>(444) 444-4040</cellPhone>
    <businessPhone>(737) 393-3838</businessPhone>
    <fax>(373) 939-3939</fax>
    <personId>1239997654</personId>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

**Figure Customer HTML Rendered (Output Data for Zone)**

### UI Map Fragments

A UI map may represent a complete HTML map or a fragment that can be embedded in a complete map.

You may use a UI map fragment in the following use cases:

- To share and reuse the same HTML/JavaScript across multiple complete UI maps. The UI map fragment is referenced as part of the HTML definition of the complete UI maps.
- To embed complex HTML/JavaScript logic as part of a map that its HTML is otherwise rendered using UI Hints.
- To implement a filter map for a data explorer search zone where basic filter definition syntax does not handle your business requirements.

### Configuration Tool Support

Configuration tool support for UI Maps hinges around the ability to inject and extract an XML document from the HTML. For more information on the specialized support for HTML and JavaScript functionality refer to UI Map Attributes and Functions.

- Ensuring Unique Element IDs for UI Maps

## Ensuring Unique Element IDs for UI Maps

The following describes how to modify JavaScript code to ensure the proper rendering of unique element IDs for UI Maps.

The modification is required only for code that renders HTML using a getElementById() (or similar) function to generate list IDs and avoid account verification or related errors.

The following sample snippet contains the necessary modifications:

```
...
function getElementsFromList(namePrefix) {
    var ret = [];
    var elements = document.getElementsByTagName("INPUT");
    for(var i=0;i<elements.length;i++) {
        var elemID = elements[i].id;
        if((id) && (id.startsWith(namePrefix + '_')) {
            ret.push(elements[i]);
```


Copyright © 2026, Oracle and/or its affiliates.

```
        }
    }
...
return ret;
```

Since IDs aren't necessarily unique in generated UI Map IDs, the code shown above ensures uniqueness at runtime by appending an underscore and row number (e.g., myField_1, myField_2) for proper handling by Framework in the rendered HTML, while still allowing you to reference the unmodified IDs contained in the generated UI Map.

A switch in the spl.properties file also permits you to disable the generation of unique IDs for elements in a grid (as described below), though, for standards compliance reasons, it is highly recommended that this switch be left at is default value.

```
Property Name: spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs
File Name: spl.properties (under web project in FW)
Default Value: false
Accepted Values: true or false
Description: This property controls the generation of unique IDs for all
input elements inside
a list. When this value is set to true it disables the generation of unique
IDs, thus
replicating the old behavior. When this property is set to false or this
property is missing
it enables the generation of unique IDs, thus enabling the list to be
standards-compliant.
```

# Understanding UI Hints

A schema may be used to describe the structure of a business object or any other XML document. When applicable, the same schema may also embed user interface (UI) "hints" that are interpreted at run time and render the HTML needed to display or maintain that document. Refer to UI Hint Syntax for more information.

The main advantages of this technique are as follows:

- **Less error prone.** A tighter correlation between the data structure and its presentation layer keeps the latter up to date, for example:
  - Adding an element to the schema or removing an element from it result in a corresponding change to the presentation layer. Note that you can explicitly exclude elements from the presentation layer as needed.
  - The HTML widget that is most appropriate to present or input data for an element is determined by its data type as defined by the schema. For example, a date picker is used for an input date element. If the element's data type changes in the schema its presentation adjusts accordingly.
- **Faster to implement.** No HTML skills are needed. The schema definition along with its embedded UI hint instructions is used to render the HTML at run time. Eliminating the need of crafting HTML and maintaining it as a UI Map reduces implementation and maintenance costs.
- **Supports business object inheritance.** Incorporating the presentation layer into the business object schema allows each business object in a hierarchy to contribute their part to the complete HTML that displays the entity. In other words, the complete HTML used to


Copyright © 2026, Oracle and/or its affiliates.

display a subclassed business object is made of the HTML rendered by the included schema of its parent business object, followed by elements extended by the subclass.

The topics in this section describe concepts and guidance relevant to the UI Hints functionality.

### Display vs. Input Modes

The same schema may be used to generate either a display version of its HTML map or an input map. In the display version, element values are presented as read-only, corresponding descriptions are shown instead of codes, and additional artifacts such as links are available for navigation to related pages. In the input version, the element is editable, using the appropriate HTML widget for its data type. For example, a foreign key element is rendered along with search components, and a date element incudes a date picker widget.

The type of map generated depends on the context in which it is used:

- The display map is generated when the schema is used to present content in a map zone.
- The input map is generated when the schema is used for data entry as part of a BPA script step or a process flow panel. Refer to Generate Edit Map Statements and process flows for more information.

By default, each element in the schema is assumed to be included in both the display and input presentation maps. UI hints allow an element to be excluded from presentation altogether, included only in the display map or included only in the input map.

### Not Just Business Object Schemas

While mostly used to describe the display and maintenance maps of business objects, UI Hints may also be embedded as part of other schema based objects:

- You may use a data area to describe an input form and prompt the user to fill it in using a BPA script step or a process flow panel.
- You may embed UI hints in a business service or service script schema for the purpose of displaying the content prepared by the service in a portal zone. Refer to map zones for more information.
- You may embed UI hints in a service script's schema to prompt for user input before calling the script to perform a task and presenting results. Refer to Generate Edit Map Statements for more information.

### Definition Order is Presentation Order

Elements are presented in the order they are defined in the expanded schema. For example, a subclassed business object that includes its parent business object followed by extension elements is presented with the parent portion of the HTML, followed by the extension fields.

### Use UI Map Fragments for Complex HTML

The following items describe basic capabilities supported by the UI Hints functionality:

- Control which elements are presented in the display and input maps.
- Provide a title for the input map.
- Introduce sections around elements by injecting start and end section instructions. The order in which the elements are defined within a section is also their presentation order.
- Basic visibility rules are supported at the element and section level.

For more complex requirements around a specific element, you may craft the necessary HTML/JavaScript for the element, capture it as a UI Map Fragment, embed the fragment as


Copyright © 2026, Oracle and/or its affiliates.

part of the schema in the appropriate presentation order, and suppress the element from basic rendering for the corresponding input or display map.

> **Note**
>
> You may also use this technique to display elements in a different presentation location. For example, when a business object's display script needs to display additional elements that are not part of the business object's schema. In this situation, the script's schema introduces the elements as suppressed and the business object schema includes an embedded HTML UI map fragment to display them in the appropriate section. Note that this may only work when both the business object and script are owned by your organization.

Refer to UI Hint Syntax for more information.

# UI Hint Syntax

### Working Examples

For working examples of uiHint functionality, refer to the following business objects:

#### BOs with User Assigned Keys

The following examples illustrate the patterns used to enable uiHints on an object with a user specified key.

- `F1-OutcomeStyleLookup`. This extendable lookup BO does not require state transition, but does allow duplicate and delete actions.
- `F1-TodoSumEmailTyp`. This request type illustrates the hints required to support state transition on a display map.
- `F1-WebSvc`. This web service BO is a good example for management of complex JavaScript requirements. Both display and input maps have functionality that requires specialized javascript.

#### BO with System Generated Key

The following example illustrates the pattern used to enable uiHints on an object with a system generated key.

- `F1-GenericAttachment`. This attachment BO has a system assigned key, which entails the following special handling:
  - `F1-AttachmentMain`. This is the main section data area contains the elements common to all attachments, including the key, bo, and version. Because this data area is used to define the main section of the generated maps, the main section of the map can be extended by an implementation via data area extension functionality.
  - `F1-AttachmentActions`. This record actions map contains the standard actions, Edit and Delete, plus custom actions used only by attachments, View and Upload.
  - `F1-AttachmentIDFrag`. This record information map contains the primary key of the attachment.

#### Display Map Service Script

Display map service scripts can be fully supported via dynamic HTML generation. However, to help eliminate the need for a display service script, self-contained uiHint functionality has been


Copyright © 2026, Oracle and/or its affiliates.

developed to write the business object status and determine valid state transitions. So the two most common reasons to craft a display service script have been eliminated.

A typical reason to use a display pre-script is if you have an embedded map fragment that contains a business service schema. The display service script can be used to invoke the business service. Both the map fragment and the display service script must declare the business service schema to support this scenario.

> **Warning**
>
> The zone used to display the object's map must have a derivation script, like `F1-GncDsMpDZ` or `F1-GenDss`, that will invoke a display service script for the business object if it has been defined as a BO option - but not require an explicit display map BO option. In addition, the display service script's schema must be enabled for uiHint functionality - as the script's schema will be dynamically rendered by the zone - and not the BO schema.

- `F1-ExcelSpreadsheet`. This attachment BO has a display service script used to manipulate the attachment business object before displaying it:
- `F1-AttchDtlU`. This display map service script's schema has been defined with the uiHint namespace, and will have a display map generated for it.

**Maintenance Pre-Processing Service Script**

Maintenance pre-processing service scripts can be used with uiHints.

- `F1-ExcelSpreadsheet`. This attachment BO has a maintenance pre-processing service script used to manipulate the attachment business object before rendering the maintenance map:
- `F1-AttchPre`. This pre-processing service script's schema mimics a maintenance map schema with embedded boGroup and action elements. It will be invoked before the maintenance map is rendered.

**Maintenance Post-Processing Service Script**

Maintenance post-processing service scripts can be used with uiHints.

- `F1-ExcelSpreadsheet`. This attachment BO has a maintenance post-processing service script used to manipulate the attachment business object after rendering the maintenance map:
- `F1-AttchPost`. This post-processing service script's schema mimics a maintenance map schema with embedded boGroup and action elements. It will be invoked after the maintenance map is rendered.

### Technical Notes

The following prerequisites are required to support dynamic HTML generation:

### Schema Requirements

To support automated UI generation, the business object schema must contain the following:

- <schema **xmlns:uiHint="http://oracle.com/ouafUIHints"**>. The schema node must name the uiHint namespace.
- **isPrimeKey="true".** Every element of the business object schema that is part of the primary key must be identified.


Copyright © 2026, Oracle and/or its affiliates.

### Maintenance Script Requirements

The maintenance script for the MO must be enabled for dynamic generation.

> **⚠ Caution**
>
> The business object maintenance BPA script must be declared as an MO Option for uiHint maintenance functionality to work!

If the script performs `F1-BOProc` then it is likely no special functionality is needed. However, if the maintenance script contains its own call to `F1-GetValOpt` then the following statement is required prior to that call:

```
move 'false' to "F1-GetBOOpts/input/maintenanceMapRequired";
performScript 'F1-GetValOpt';
```

After the call to `F1-GetValOpt` the following logic must be included to dynamically declare the map schema if the business object does not have a maintenance map of its own:

```
 // Perform Main Processing
if ("F1-GetBOOpts/output/maintenanceMap = $BLANK")
  declareBOWithBOGroup "$bo" as 'map_schema';
else
  declareMap "F1-GetBOOpts/output/maintenanceMap" as 'map_schema';
end-if;
```

### Format an Input Map Title

> **ⓘ Note**
>
> Throughout this topic the term "field" to refer to both the generic concept of displaying and capturing data in a 'field' as well as referring to the meta-data object supplied in the product to define Fields. When referring to the latter, the term "MD Field" (meta-data Field) is used.

A uiHint element can be used to build a title for a maintenance map. The title will only print on the maintenance map, not on the display map. It will be printed as the first line in the map, centered, with a heading style.

| Syntax | Description | Examples |
|---|---|---|
| `<uiHint:title mdField=""/>` | Displays the label of a referenced MD field as the title. | `<schema xmlns:uiHint="http://oracle.com/ouafUIHints"> <uiHint:title mdField="STATUS_RSN_LBL"/> ... </schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Description | Examples |
|---|---|---|
| `<uiHint:title text=" "/>` | Displays the indicated text as the title. (Do not use this mechanism when multiple languages are supported. | `<schema xmlns:uiHint="http://oracle.com/ouafUIHints"> <uiHint:title text="Status Reason"/> ... </schema>` |

### Create a Section

The uiHint namespace supports the definition of a UI map section. Note that sections are currently created in generated UI Maps when the schema has a group or list node with a label or mdField. The functionality described here enables the creation of a section without requiring a labeled group or list node within the schema. Every section must be bounded by **startSection** and **endSection** element pair.

| Syntax | Supporting Attributes | Description |
|---|---|---|
| `<uiHint:startSection .../>` | `sectionColumn="left \| right \| fullWidth \| float"` | The default is that the section will be the full width in display maps. To override that setting, specify if you want a half-width section to appear in either the left (**left**) or right (**right**) column or to float (**float**). Sections that are marked as 'float' will display half-width and be aligned according to whether prior sections are displayed or conditionally hidden. For example, if a left-aligned section is followed by a floating section, the floating section will appear in the right column if the left section is populated but will display in the left column if the left section is hidden / collapsed. |
| | `editColumn="left \| right \| fullWidth \| float"` | By default a section appears as full width in maintenance maps. To override that setting, specify if you want a half-width section to appear in either the left (**left**) or right (**right**) column or to float (**float**). The behavior is the analogous to the **sectionColumn** behavior. |
| | `sectionOpen="false"` | By default a section is open on initial display. Specify this attribute to initially display the section as closed (collapsed). |
| | `mdField=" "` | Specify the name of a MD field whose label should be used as the section heading. |
| | `label=" "` | Specify the explicit text to use as the section heading. |
| | `visibleOn="displayMap \| inputMap"` | By default a section appears on both the display and the input maps. Use this attribute to limit the display of the section to either the display map (**displayMap**) or input map (**inputMap**). |

The syntax for the end section attribute is `<uiHint:endSection/>`


Copyright © 2026, Oracle and/or its affiliates.

Examples:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    <uiHint:startSection label="Main" sectionColumn="left"/>
    ...
    <uiHint:endSection/>
</schema>


<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    <uiHint:startSection mdField="F1-ADD-INFO" sectionColumn="fullWidth"
editColumn="float" sectionOpen="false" visibleOn="displayMap"/>
    ...
    <uiHint:endSection/>
</schema>
```

> **Note**
>
> The sectionColumn, editColumn and sectionOpen attributes are available for group and list nodes as well.

### Include a Map Fragment

You can specify a UI map fragment to inject HTML into a generated map using the **includeMap** element name. This allows for you to support more sophisticated behavior on your user interface. For any element that is included for rendering in the map fragment, be sure to suppress the element in its schema definition, otherwise HTML will automatically be generated for the element.

| Syntax | Supporting Attributes | Description |
|---|---|---|
| `<uiHint:includeMap ../>` | `map=" "` | Specify the name of the map. |
| | `visibleOn="displayMap \| inputMap"` | By default the details from the map fragment appear on both the display and the input maps. Use this attribute to limit the display of the section. |

Example:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
  ...
  <uiHint:includeMap map="StandardActionButtons" visibleOn="displayMap"/>
  ...
</schema>
```


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Important note on the map fragment schema:** If a map fragment contains a schema, then the fragment schema structure will be injected into the dynamically generated schema when the business object is rendered for input. Technically, the fragment schema will be inserted after the boGroup structure within the map's schema. This method may be used to support the implementation of maintenance pre and post script processing for a business object and oraInvokeBS function calls within embedded JavaScript.

If JavaScript is required within an XHTML UI Map fragment, it is necessary to bound it within a **![CDATA[ ]]** tag to ensure a valid XML document. Note that the tags themselves may need to be commented out to promote compatibility with older browsers. For example:

```
<script type="text/javascript">
/* <![CDATA[ */
//
//javascript
//
/* ]]> */
</script>
```

**Flush the cache**: For performance reasons, the Framework automatically caches business object schemas, data areas, and UI maps. When you update a business object, the cache is automatically flushed. However, if the business object includes either a data area or embedded UI map fragment, the cache must be manually flushed in order for your changes to be recognized. Refer to Server Cache for more information.

### Build A Dropdown

Syntax is provided to build a dropdown list in an edit map. The dropdown may be built using data returned from a service script, a business service or a table.

| Syntax | Description |
|---|---|
| `uiHint:select="ss: "` | Specify the name of the service script after the colon. |
| `uiHint:select="bs: "` | Specify the name of the business service after the colon. |
| `uiHint:select="table: "` | Specify the name of the table after the colon. |

When specifying a service script or a business service, extra mapping information is needed to pass data to and from the service.

| Syntax | Values | Description |
|---|---|---|
| `uiHint:selectIn=" "` | serviceXPath:element | Used to pass the value of another element into the service (mapping to the service's XPath). |
| | serviceXPath:'Literal' | Used to pass a constant or literal to the service (mapping to the service's XPath). |


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Values | Description |
|---|---|---|
| `uiHint:selectOut="valuePath: ; descPath: "` | See examples below. | Used to indicate which element in the service's output holds the values and which one holds the descriptions. |

Examples:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    <boStatus mapField="BO_STATUS_CD" uiHint:select="bs:F1-BOStateReasonList"
    uiHint:selectIn="boStatusBO:boStatusBO"
uiHint:selectOut="valuePath:results/status;
    descPath:results/description"/>
    ...
    <algorithm mdField="ALG_CD" uiHint:select="bs:F1-
RetrieveSysEvtAlgorithms"
     uiHint:selectIn="algorithmEntity:'F1AA';"
uiHint:selectOut="valuePath:results/algorithm;
     descPath:results/description"/>
    ...
    <outboundMsgType mdField="OUTMSG_TYPE_CD" required="true" fkRef="F1-
OMTYP" uiHint:select="table:F1_OUTMSG_TYPE"/>
</schema>
```

### Conditionally Hide Elements

The **displayNone** attribute is used to suppress elements on the map based on conditions.

| Syntax | Values | Description |
|---|---|---|
| `uiHint:displayNone=` | "'XPath','value','!=' \| '='" | Used to conditionally hide this element based on the value of another element (referenced using its XPath). Enter a value of `' '` to interrogate a blank value. By default the operator is `'='`. This may be overridden using `'!='`. |
| | "function name, **true** \| **false**" | Used to indicate a JavaScript function, which must return a Boolean. |

> **Warning**
>
> Embedded spaces are not supported within the comma separated string values of this attribute.
>
> This setting may be used on group nodes, list nodes, and elements - except for elements within a list. Elements within a list cannot be hidden conditionally.

The following example illustrates that two elements (currency reference and lookup) that will be hidden or displayed based on the value of the data type element. Note that this example also


Copyright © 2026, Oracle and/or its affiliates.

illustrates Trigger Dependent Behavior because the data type element's value may change and if it does, the condition for hiding the subsequent elements should be re-evaluated.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
    ...
    <dataType mdField="F1_SE_DATA_TYPE" dataType="lookup"
lookup="F1_SE_DATA_TYPE"
     uiHint:dependents="currencyRef;lookup; "/>
     <currencyRef mdField="F1_SE_CURR_REF_LBL"
uiHint:displayNone="'dataType','F1MO','!='"/>
     <lookup mdField="F1_SE_LOOKUP_LBL" fkRef="F1-LKUPF"
uiHint:displayNone="'dataType','F1LP','!='"/>
     ...
</schema>
```

The following example illustrates referring to a function where the function receives parameters:

```
<uiHint:startSection mdField="F1_SE_DEFAULT_SECT"
    uiHint:displayNone="isApplicableForSchemaType(item,'F1MP'),true"/>
```

### Conditionally Protect Elements

The protect attribute is used to protect elements on the map based on other factors.

| Syntax | Values | Description |
|---|---|---|
| `uiHint:protect=` | "'XPath','value','!=' \| '='" | Used to conditionally protect this element based on the value of another element (referenced using its XPath). Enter a value of ' ' to interrogate a blank value. By default the operator is `'='`. This may be overridden using `'!='`. |
| | "function name, **true** \| **false**" | Used to indicate a JavaScript function, which must return a Boolean. |
| | "'**action**','A' \| 'C','!=' \| '='" | Use the `'action'` setting to protect the element based on the current action. For example, certain elements may only be specified when adding a record. Any subsequent changes to the record should protect the element from being changed. When using this option, the valid values for the *'value'* are `A` (add) and `C` (change). |


Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> Embedded spaces are not supported within the comma separated string values of this attribute.
>
> The protect UI Hint may be used on group nodes, list nodes, and elements - except for elements within a list. Elements within a list cannot be protected conditionally.

The following UI Hint will protect the statistics category when the action is 'C'.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
     ...
     <statisticsCategory dataType="lookup" mapField="STAT_CATEGORY_FLG"
    lookup="STAT_CATEGORY_FLG" uiHint:protect="'action','C','='"/>
     ...
    </schema>
```

### Trigger Dependent Behavior

The dependents attribute is used to trigger behavior on a child element when a parent element is changed.

| Syntax | Values |
|---|---|
| `uiHint:dependents=" "` | A list of one or more dependent elements separated by semicolons. |

The following example illustrates that the dropdown list of one element is driven by the value of another element. In this example, when the Country changes, the list of States to choose from should change to only show the states for the indicated country.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
  <country label="Country" uiHint:select="table:CI_COUNTRY"
uiHint:dependents="state"/>
  <state label="State" uiHint:select="ss:CM-RetrieveCountryStates"
   uiHint:selectIn="input/country:country;"
uiHint:selectOut="valuePath:output/state/stateCode;
   descPath:output/state/stateDesc"/>
    ...
</schema>
```

> **Note**
>
> Dependent targets may only name elements, not group or list nodes.
>
> Do not modify the "id" attribute value of dependent and parent element. Data population in dependent is done based on the "id" attribute value.


Copyright © 2026, Oracle and/or its affiliates.

### Control Rendering Target

By default all elements that are not suppressed are visible on both the display map and the input map. Use the **visibleOn** attribute to limit the inclusion of an element to either the display or input map.

| Syntax | Values |
|---|---|
| `uiHint:visibleOn=` | `"displayMap"`<br>`"inputMap"` |

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
  ...
  <uiHint:includeMap map="StandardActionButtons" visibleOn="displayMap"
  ...
</schema>
```

### Generate a Text Area

By default, a standard text box is rendered in an input map for any string element. If the field is larger and you wish to have a bigger text area (with a scroll bar), use the **textArea** attribute.

| Syntax |
|---|
| `uiHint:textArea="true"` |

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
   ...
  <message label="Message" uiHint:textArea="true"/>
  ...
</schema>
```

### Modify FK Reference Defaults

By default, when an element with **fkRef** is displayed, an info string, context menu, navigation, and search are enabled (if the FK reference has been configured accordingly). Syntax is provided to allow you to selectively turn off any of these features.

| Syntax |
|---|
| `uiHint:fkRef="info:false;context:false;navigation:false;search:false;"` |

Only the feature that you wish to turn off needs to be specified. The following example illustrates turning off the navigation capability, meaning the text will not be rendered as hypertext.

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
   ...
   <attachmentID fkRef="F1-ATTCH" primeKey="true" suppress="input"
uiHint:fkRef="navigation:false;"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   ...
</schema>
```

> **NOT_SUPPORTED**
>
> Refer to FK Reference Formatting in the UI Map Attributes section for more information on each FK reference setting.

### Suppress Automatic Number Formatting

By default numeric fields (`dataType="number"`) are formatted as numbers. An attribute is provided to instead apply alphanumeric formatting.

Note: If **dataType** is not specified explicitly, it is be derived from **mdField** or **mapField**.

| Syntax |
| --- |
| `uiHint:alphaFormat="true\|false"` |

By default, its value is **false** (and therefore can be left out altogether).

Examples:

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
     ...
     <numberCount mdField="" dataType="number" uiHint:alphaFormat="true"/>
        ...
    </schema>
```

### Auto Capitalize the Input Data

The uiHint provides syntax to automatically capitalize input data.

| Syntax |
| --- |
| `uiHint:capitalize="true\|false"` |

By default, its value is **false** (and therefore can be left out altogether).

```
<schema xmlns:uiHint="http://oracle.com/ouafUIHints">
     <toDoTypeCd mdField="TD_TYPE_CD" uiHint:capitalize='true'
isPrimeKey="true"/>
</schema>
```

> **Note**
>
> This attribute is ignored if `uiHint:textArea="true"` is configured.

The attribute is only available in the schema designer when the **isPrimeKey** is set to **true**. The attribute may be added to any string element when using the source viewer.


Copyright © 2026, Oracle and/or its affiliates.

### Override Row Header for Accessibility Purposes

When generating the HTML for a list, the default is to assign scope="row" to the non-suppressed primary key elements, if any, or the first non-suppressed element in the list. This uiHint provides syntax to override the defaults when they are not the appropriate elements to be used by a screen reader.

| Syntax |
| --- |
| `uiHint:rowHeader="true\|false"` |

```
<iwsAnnotationTypeParameter mdField="F1_ANN_TYPE_PARMS_LBL" type="list"
mapChild="F1_IWS_ANN_TYPE_PARM">
        <sequence dataType="number" mapField="SEQ_NUM"
uiHint:rowHeader="false"/>
        <parameterName mapField="PARM_NAME" uiHint:rowHeader="true"
dataType="string"/>
        <parameterValue mapField="PARM_VALUE" dataType="string"/>
        <isRequired dataType="lookupBO" mapField="REQUIRED_SW" lookupBO="F1-
BooleanValues"/>
        <javaPackage mapField="JAVA_PACKAGE" dataType="string"/>
        <version suppress="true" dataType="number" mapField="VERSION"/>
        <description mapField="DESCR" dataType="string"/>
        <longDescription mapField="DESCRLONG" uiHint:textArea="true"
dataType="string"/>
        <owner suppress="input" dataType="lookup" mapField="OWNER_FLG"
lookup="OWNER_FLG"/>
    </iwsAnnotationTypeParameter>
```

The attribute is only applicable to elements in a list.

# Defining UI Maps

Use this page to define basic information about a user interface (UI) Map.

You can access the portal from the **Admin Menu**, and then **System**, and then **UI Map**. You are brought to a query portal with options for searching for a specific UI Map. Once a UI Map has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a new UI Map, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

Refer to Understanding UI Maps for more information.

Use **UI Map Type** to indicate the map type. The system recommends using a **Complete XHTML Document** (the default) or **XHTML Fragment**. The values **Complete HTML Document** and **HTML Fragment** remain for backward compatibility but are not recommended.

The following zones may appear as part of the portal's **Main** tab page.


Copyright © 2026, Oracle and/or its affiliates.

- **UI Map.** This zone provides information about the selected UI Map.
- **HTML Editor.** This zone allows you to edit the HTML document of the map.

> **Note**
>
> Refer to UI Map Attributes and Functions and UI Map Standards for more information about HTML definition syntax. These topics describe good ways to produce simple HTML, however, they are not an HTML reference. Note that you can use a variety of HTML editors to compose your HTML, which you can then cut and paste into your UI map.

> **Note**
>
> **Providing Help.** A tool tip can be used to display additional help information to the user. This applies to section elements as well as individual elements on a map. Refer to UI Map Attributes and Functions for more information on how to enable and provide UI map help.

- **HTML Override Editor.** This zone allows you to provide an override HTML document for a base product map.

The following zones may appear as part of the portal's **Schema** tab page.

- **Schema Editor.** This zone allows you to edit the UI Map's schema using the Schema Designer tool. The purpose of the schema is to describe the UI Map's properties.

> **NOT_SUPPORTED**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

> **Tip**
>
> Click the **View Schema** hyperlink to view the UI Map's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **References** tab page.

- **UI Map References.** This zone list various configuration entities referencing the current UI Map, its HTML or its schema.

### Actions

The page supports the following special actions:

- **View Schema.** Click the hyperlink to view the UI Map's expanded schema definition. Doing this opens the schema viewer window.
- **Test UI Map.** Use the hyperlink to render your HTML in a test window. This action also exercises the proprietary functionality that binds an XML element with an HTML element so you can get immediate feedback on your HTML syntax.
- UI Map Attributes and Functions


Copyright © 2026, Oracle and/or its affiliates.

- UI Map Standards

# UI Map Attributes and Functions

> **Note**
>
> Use the UI Map Query's Additional Information option to review HTML examples provided by the product.

### Bind XML to HTML

Only two different attributes are required to bind a UI Map's XML to its HTML. Both of these attributes require an XML document embedded within the HTML, where the XML is bounded by <xml> nodes.

> **Warning**
>
> You must embed a pair of <xml></xml> tags within your HTML document for binding to occur.

### Linking a Field

| Syntax | Values | Description |
|---|---|---|
| oraField=" " | Field element XPath | This attribute is used to link an HTML element directly with an XML element, where the XML element is defined within the UI Map's XML schema. The attribute can be used with any *rendering* HTML element, such as: , , and <input>. |

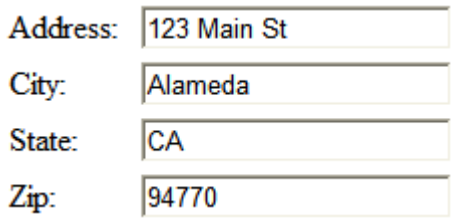
- HTML for input element:

```
<html>
<body>
<table>
   <tr>
      <td>Address:</td>
      <td><input type="text" oraField="address"/></td>
   </tr>
   <tr>
      <td>City:</td>
      <td><input type="text" oraField="city"/></td>
   </tr>
   <tr>
      <td>State:</td>
      <td><input type="text" oraField="state"/></td>
   </tr>
   <tr>
      <td>Zip:</td>
      <td><input type="text" oraField="zip"/></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
</table>
</body>
<xml>
   <root>
      <address>123 Main St</address>
      <city>Alameda</city>
      <state>CA</state>
      <zip>94770</zip>
   </root>
</xml>
</html>
```

Rendered HTML




- HTML for span and div elements:

```
<html>
<body>

<div oraField="address"></div>
<span oraField="city"></span>
<span>,</span>
<span oraField="state"></span>
<span oraField="zip"></span>
<span oraField="country"></span>

</body>
<xml>
   <root>
      <address>123 Main St</address>
      <city>Alameda</city>
      <state>CA</state>
      <zip>94770</zip>
   </root>
</xml>
</html>
```

HTML rendered:

123 Main St
Alameda, CA 94770

**Linking a List**


Copyright © 2026, Oracle and/or its affiliates.

This attribute is used to link an HTML table with an XML list, where the XML list is defined within the UI Map's XML schema. The purpose of the element is to trigger the framework to replicate the table's HTML for each occurrence of the list.

| Syntax | Values | Description |
|---|---|---|
| oraList=" " | List element XPath | This attribute is used to link an HTML table with an XML list, where the XML list is defined within the UI Map's XML schema. The purpose of the element is to trigger the framework to replicate the table's HTML for each occurrence of the list. |

> **Note**
>
> The oraField attributes embedded within the list must contain XPath navigation relative to the list. See below for an example.

```
<html>
<head><title>Bind xml list element</title></head>
<body>
<table oraList="payment">
   <thead>
      <tr>
         <th><span>Pay Date</span></th>
         <th><span>Amount</span></th>
        </tr>
        <tr/>
    </thead>
    <tr>
      <td>
         <span oraField="date" oraType="date"></span>
      </td>
      <td align="right">
         <span oraField="amount" oraType="money"></span>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
    <payment>
        <date>2008-01-01</date>
        <amount>44.28</amount>
    </payment>
    <payment>
        <date>2008-02-01</date>
        <amount>32.87</amount>
    </payment>
    <payment>
        <date>2008-03-01</date>
        <amount>21.76</amount>
    </payment>
</root>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xml>
</html>
```

HTML rendered:

| Pay Date | Amount |
|---|---|
| 01-01-2008 | $44.28 |
| 02-01-2008 | $32.87 |
| 03-01-2008 | $21.76 |

### Build a Dropdown List

The following attributes are provided to build an HTML ‘select’ element, also called a dropdown, based on various sources.

| Source | Syntax | Values | Examples |
|---|---|---|---|
| Lookup | `oraSelect="lookup: ;"` | Lookup field | `...`<br>`<td>House Type:</td>`<br>`<td>`<br>`<select oraField="houseType" oraSelect="lookup:HOUSE_TYPE;" >`<br>`</select>`<br>`</td>` |
| Extendable Lookup | `oraSelect="lookupBO: ;"` | BO code | `...`<br>`<td>UI Device Display Type:</td>`<br>`<td>`<br>`<select oraField="uiDeviceType" oraSelect="lookupBO: F1-DeviceDisplayTypes;"></select>`<br>`</td>` |
| Characteristic Type (pre-defined) | `oraSelect="charType: ;"` | Characteristic Type code | `...`<br>`<td>Usage:</td>`<br>`<td>`<br>`<select oraField="statusReasonUsage" oraSelect="charType:F1-SRUSG;"></select>`<br>`</td>`<br>`...` |


Copyright © 2026, Oracle and/or its affiliates.

| Source | Syntax | Values | Examples |
| --- | --- | --- | --- |
| Control Table | `oraSelect="table: ;"` | Table name | ```<br>...<br><td>Currency: </td><br><td><br><select<br>oraField="currency"<br>oraSelect="table:CI_CURRENCY_CD;"><br></select><br></td><br>...<br>``` |

**Note**

This attribute only works with tables that follow the esta

| Source | Syntax | Values | Examples |
| --- | --- | --- | --- |
| | | ndardcontroltablestructurewhereretherereisarelatedlanguageta | |


Copyright © 2026, Oracle and/or its affiliates.

| Source | Syntax | Values | Examples |
|---|---|---|---|
| | | b<br>l<br>e<br>t<br>h<br>a<br>t<br>i<br>n<br>c<br>l<br>u<br>d<br>e<br>s<br>t<br>h<br>a<br>t<br>i<br>n<br>c<br>l<br>u<br>d<br>e<br>s<br>t<br>h<br>e<br>c<br>o<br>l<br>u<br>m<br>n<br>**D**<br>**E**<br>**S**<br>**C**<br>**R**<br>o<br>r<br>D<br>E<br>S<br>C<br>R<br>1<br>0<br>0<br>a<br>s<br>i<br>t | |


Copyright © 2026, Oracle and/or its affiliates.

| Source | Syntax | Values | Examples |
| --- | --- | --- | --- |
| | | sdescriptioncolumn. Go to the Field maintenance page for each field | |


Copyright © 2026, Oracle and/or its affiliates.

| Source | Syntax | Values | Examples |
| --- | --- | --- | --- |

dtoseewhattablescontainthefields.

**Warning**

TheoraSelect

| Source | Syntax | Values | Examples |
| --- | --- | --- | --- |
| | | function will only work if less than 500 values are displayed. | |


Copyright © 2026, Oracle and/or its affiliates.

| Source | Syntax | Values | Examples |
|---|---|---|---|
| Page Service | `oraSelect="service: ;"` | Page Service name | `...`<br>`<td>Country:</td>`<br>`<td>`<br>`<select oraField="country" oraSelect="service:CIPTCNTW;">`<br>`</select>`<br>`</td>`<br>`...` |
| Embedded List | Used to build a select dropdown based on a list within the map's XML. `oraSelect="valuePath: ;descPath: "` | After the valuePath, indicate the XPath of the element that holds the values. After the descPath, indicate the XPath of the element that holds the descriptions. | `<html>`<br>`<body>`<br>`<table summary="" border="0" cellpadding="1" cellspacing="1">`<br>`<tr>`<br>`<td>Select: </td>`<br>`<td><select oraSelect= "valuePath:list/value; descPath:list/desc" oraField="target"></ select></td>`<br>`</tr>`<br>`</table>`<br>`</body>`<br>`<xml>`<br>`<root>`<br>`<target>10</target>`<br>`<list>`<br>`<value>10</value>`<br>`<desc>Ten</desc>`<br>`</list>`<br>`<list>`<br>`<value>20</value>`<br>`<desc>Twenty</desc>`<br>`</list>`<br>`<list>`<br>`<value>40</value>`<br>`<desc>Forty</desc>`<br>`</list>`<br>`</root>`<br>`</xml>`<br>`</html>` |
| Service Script | `oraSelect="ss: ; "` | Service Script code | See below for additional syntax needed when using this function. |
| Business Service | `oraSelect="bs: ; "` | Business Service code | See below for additional syntax needed when using this function. |


Copyright © 2026, Oracle and/or its affiliates.

When specifying a service script or a business service, extra mapping information is needed to pass data to and from the service.

| Syntax | Values | Description |
|---|---|---|
| oraSelectIn=" ;" | serviceXPath:element; | Used to pass the value of another element into the service (mapping to the service's XPath). |
| | serviceXPath:'Literal'; | Used to pass a constant or literal to the service (mapping to the service's XPath). |
| oraSelectOut= "valuePath: ; descPath: " | See examples below | Used to indicate which element in the service's output holds the values and which one holds the descriptions. |

Example using a business service:

```
...
<td>External System: </td>
<td>
   <select oraField="externalSystem"
    oraSelect="bs:F1-RetrieveExternalSystems"
    oraSelectIn="outboundMsgType:boGroup/parameters/outboundMsgType;"
    oraSelectOut="valuePath:results/externalSystem;
    descPath:results/description">
   </select>
</td>
...
```

This method for building dropdowns is often used when there is a dependency between elements and the list of valid values in a dropdown (for the child element) is based on another element in the map (the parent element). When the parent element is changed, it may be required to refresh the child element. This behavior can be implemented using the function called within an **onChange** event in the map. The syntax is **oraHandleDependentElements('dependent element');**. Multiple target elements (dependents) can be named.

The following example is related to the above business service example where the list of external systems is specific for a given outbound message type, which is passed in as input. The snippet below shows the configuration for the outbound message type element to trigger a refresh of the external system’s dropdown list.

```
...
<div>
   <label oraLabel="boGroup/parameters/outboundMsgType"></label>
      <span>
         <select oraSelect="table:F1_OUTMSG_TYPE"
          oraField="boGroup/parameters/outboundMsgType"
          onChange="oraHandleDependentElements('boGroup/parameters/
externalSystem');">
         </select>
      </span>
</div>
...
```


Copyright © 2026, Oracle and/or its affiliates.

### Format Input and Output Fields

The following attributes are designed to apply data type formatting for input and output fields.

#### Automatic Formatting

| Syntax |
| --- |
| `oraSchemaDataTypes="false"` |

This attribute is used to trigger automatic formatting in the rendered HTML document. The automated formatting will occur according to the data type attributes defined in the UI map's schema. For details on specific data type formatting, please refer to the oraType attribute descriptions below.

> **Warning**
>
> The attribute **oraSchemaDataTypes="true"** will be automatically injected into the UI map's HTML! If you do not wish to apply the schema's data types to the rendered HTML then you must specify this attribute in the body node with a value of false. The attribute **<body oraSchemaDataTypes="false">** is required to avoid automatic formatting!

- UI Map schema:

```
<schema>
    <schemaDate dataType="date"/>
    <schemaDateTime dataType="dateTime"/>
    <schemaFkRef fkRef="F1-MOTBL"/>
    <schemaLookup dataType="lookup" lookup="ACCESS_MODE"/>
    <schemaMoney dataType="money"/>
    <schemaNumber dataType="number"/>
    <schemaTime dataType="time"/>
</schema>
```

- UI Map HTML:

```
<html>
<body oraSchemaDataTypes="true">
<table border="1" cellpadding="1" cellspacing="1">
   <tr><th>dataType</th><th>input result</th><th> display-only result</
th></tr>
   <tr>
      <td>date</td>
      <td><input oraField="schemaDate"/></td>
      <td><span oraField="schemaDate"></span></td>
   </tr>
   <tr>
      <td>dateTime</td>
      <td><input oraField="schemaDateTime"/></td>
      <td><span oraField="schemaDateTime"></span></td>
   </tr>
   <tr>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      <td>fkRef</td>
      <td><input oraField="schemaFkRef"/></td>
      <td><span oraField="schemaFkRef"></span></td>
   </tr>
   <tr>
      <td>lookup</td>
      <td><input oraField="schemaLookup"/></td>
      <td><span oraField="schemaLookup"></span></td>
   </tr>
   <tr>
      <td>money</td>
      <td><input oraField="schemaMoney"/></td>
      <td><span oraField="schemaMoney"></span></td>
   </tr>
   <tr>
      <td>number</td>
      <td><input oraField="schemaNumber"/></td>
      <td><span oraField="schemaNumber"></span></td>
   </tr>
   <tr>
      <td>time</td>
      <td><input oraField="schemaTime"/></td>
      <td><span oraField="schemaTime"></span></td>
   </tr>
</table>

</body>
<xml>
<root>
   <schemaDate>2007-11-02</schemaDate>
   <schemaDateTime>2007-11-02-23.45.00</schemaDateTime>
   <schemaFkRef>CI_MD_TBL</schemaFkRef>
   <schemaLookup>A</schemaLookup>
   <schemaMoney>1000000</schemaMoney>
   <schemaNumber>5661976.11548</schemaNumber>
   <schemaTime>23.45.00</schemaTime>
</root>
</xml>
</html>
```

HTML rendered.

| dataType | input result | display-only result |
|---|---|---|
| date | 11-02-2007 | 11-02-2007 |
| dateTime | 11-02-2007 11:45:00PM | 11-02-2007 11:45:00PM |
| fkRef | CI_MD_TBL Table | Table |
| lookup | A | Add |
| money | $1,000,000.00 | $1,000,000.00 |
| number | 5,661,976 | 5,661,976 |
| time | 11:45:00PM | 11:45:00PM |

### Date and Time Formatting


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Valid Values | Description |
|---|---|---|
| `oraType="date"` | | This function is used to display a date according to the user's display profile. For input fields, the setting formats the data when the user tabs out of the field. |
| `oraType="time"` | | This function is used to display a time according to the user's display profile. For input fields, the setting formats the data when the user tabs out of the field. |
| `oraType="dateTime; time:suppress"` | | This function is used to display a timestamp according to the user's display profile. If this function is used for an input element, it is broken into two pieces, one for date and one for time. Optionally, the time portion of the date time element can be suppressed using the attribute value 'time:suppress'. |
| `oraType="dateTime; stdTime:true;"` | | This function is used to render a date / time element that is stored in standard time according to the daylight savings time schedule of the 'base' time zone. The 'base' time zone is specified on the Installation table and represents the database time zone.<br>For input elements with this setting, all time entered is assumed to correspond with the daylight savings time schedule of the base time zone. If a time is entered that cannot be unambiguously translated to standard time, then the user will be required to provide a time zone label to indicate whether daylight savings time, or standard time, has been entered. |
| `oraType="dateTime; stdTimeRef:;"` | Reference an XPath after the colon. | This function is used to render a date / time element according to the daylight savings time schedule of a time zone whose XPath is referenced. Note that the time processed is assumed to have been stored in the standard time of the referenced time zone - so only daylight savings time shifting will execute - not time zone shifting. |
| `oraType="dateTime; legalTime:true;"` | | This function is used to render a date / time element that is stored in legal time of the 'base' time zone. The 'base' time zone is specified on the Installation table and represents the database time zone. |
| `oraType="dateTime; legalTimeRef: ;"` | Reference an XPath after the colon. | This function is used to render a date / time element according to the daylight savings time schedule of a time zone whose XPath is referenced. Note that the time processed is assumed to have been stored in the legal time of the referenced time zone. |
| `oraType="dateTime; … ; displayRef:;"` | Reference an XPath after the colon. | This function is used to display a date/time element that is explicitly defined as stored in standard or legal time (using either stdTime \| stdRef \| legalTime \| legalRef) from its storage time zone to the referenced time zone in legal time before display and vice versa, from the display time zone to the storage time zone on save. |

```
<html>
<body oraSchemaDataTypes="true">
<table border="1" cellpadding="1" cellspacing="1">
   <tr><th>dataType</th><th>input result</th><th> display-only result</
th></tr>
   <tr>
      <td>date</td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      <td><input oraField="schemaDate" oraType="date;"/></td>
      <td><span oraField="schemaDate" oraType="date;"></span></td>
   </tr>
   <tr>
      <td>time</td>
      <td><input oraField="schemaTime" oraType="time;"/></td>
      <td><span oraField="schemaTime" oraType="time;"></span></td>
   </tr>
   <tr>
      <td>dateTime</td>
      <td><input oraField="schemaDateTime" oraType="dateTime;"/></td>
      <td><span oraField="schemaDateTime" oraType="dateTime;"></span></td>
   </tr>
   <tr>
      <td>dateTimeSuppress</td>
      <td><input oraField="schemaDateTimeSup" oraType="dateTime;
time:suppress"/></td>
      <td><span oraField="schemaDateTimeSup" oraType="dateTime;
time:suppress"></span></td>
   </tr>
   <tr>
     <td>Base Time Zone (Standard)</td>
     <td><input oraField="schemaDateTimeStd" oraType="dateTime;
stdTime:true;"/></td>
     <td><span oraField="schemaDateTimeStd" oraType="dateTime;
stdTime:true;"></span></td>
   </tr>
   <tr>
     <td>Base Time Zone (Legal)</td>
     <td><input oraField="schemaDateTimeLgl" oraType="dateTime;
legalTime:true;"/></td>
     <td><span oraField="schemaDateTimeLgl" oraType="dateTime;
legalTime:true;"></span></td>
   </tr>
   <tr>
     <td>Referenced Time Zone (Standard)</td>
     <td><input oraField="schemaDateTimeStdRef" oraType="dateTime;
stdTimeRef:timeZone;"/></td>
     <td><span oraField="schemaDateTimeStdRef" oraType="dateTime;
stdTimeRef:timeZone;"></span></td>
   </tr>
   <tr>
     <td>Referenced Time Zone (Legal)</td>
     <td><input oraField="schemaDateTimeLglRef" oraType="dateTime;
legalTimeRef:timeZone;"/></td>
     <td><span oraField="schemaDateTimeLglRef" oraType="dateTime;
legalTimeRef:timeZone;"></span></td>
   </tr>
   <tr>
     <td>Stored in Base Time Zone (Standard) - displayed in another Time
Zone</td>
     <td><input oraField="schemaDateTimeStdDisp" oraType="dateTime;
stdTime:true; displayRef:timeZoneDisp;"/></td>
     <td><span oraField="schemaDateTimeStdDisp" oraType="dateTime;
stdTime:true; displayRef:timeZoneDisp;"></span></td>
   </tr>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   <tr>
     <td>Stored in Base Time Zone (Legal) - displayed in another Time
Zone</td>
     <td><input oraField="schemaDateTimeLglDisp" oraType="dateTime;
legalTime:true; displayRef:timeZoneDisp"/></td>
     <td><span oraField="schemaDateTimeLglDisp" oraType="dateTime;
legalTime:true; displayRef:timeZoneDisp"></span></td>
   </tr>
   <tr>
     <td>Stored in a Referenced Time Zone (Standard) - displayed in another
Time Zone</td>
     <td><input oraField="schemaDateTimeStdRefDisp" oraType="dateTime;
stdTimeRef:timeZone; displayRef:timeZoneDisp"/></td>
     <td><span oraField="schemaDateTimeStdRefDisp" oraType="dateTime;
stdTimeRef:timeZone; displayRef:timeZoneDisp"></span></td>
   </tr>
   <tr>
     <td>Stored in a Referenced Time Zone (Legal) - displayed in another Time
Zone</td>
     <td><input oraField="schemaDateTimeLglRefDisp" oraType="dateTime;
legalTimeRef:timeZone; displayRef:timeZoneDisp"/></td>
     <td><span oraField="schemaDateTimeLglRefDisp" oraType="dateTime;
legalTimeRef:timeZone; displayRef:timeZoneDisp"></span></td>
   </tr>

  </table>

</body>
<xml>
<root>
   <timeZone>US-EASTERN</timeZone>
   <timeZoneDisp>US-CENTRAL</timeZoneDisp>
   <schemaDate>2007-11-02</schemaDate>
   <schemaTime>23.45.00</schemaTime>
   <schemaDateTime>2007-11-02-23.45.00</schemaDateTime>
   <schemaDateTimeSup>2007-11-02-23.45.00</schemaDateTimeSup>
   <schemaDateTimeStd>2007-11-02-23.45.00</schemaDateTimeStd>
   <schemaDateTimeStdRef>2007-11-02-23.45.00</schemaDateTimeStdRef>
   <schemaDateTimeLgl>2007-11-02-23.45.00</schemaDateTimeLgl>
   <schemaDateTimeLglRef>2007-11-02-23.45.00</schemaDateTimeLglRef>
   <schemaDateTimeStdDisp>2007-11-02-23.45.00</schemaDateTimeStdDisp>
   <schemaDateTimeStdRefDisp>2007-11-02-23.45.00</schemaDateTimeStdRefDisp>
   <schemaDateTimeLglDisp>2007-11-02-23.45.00</schemaDateTimeLglDisp>
   <schemaDateTimeLglRefDisp>2007-11-02-23.45.00</schemaDateTimeLglRefDisp>
</root>
</xml>
</html>
```

HTML rendered.


Copyright © 2026, Oracle and/or its affiliates.

| dataType | input result | display-only result |
|---|---|---|
| date | 11-02-2009 | 11-02-2009 |
| time | 12:28:54AM | 12:28:54AM |
| dateTime | 11-02-2009 12:28:54AM | 11-02-2009 12:28:54AM |
| dateTimeSuppress | 11-02-2009 | 11-02-2009 |
| Base Time Zone (Standard) - Duplicate hour | 11-01-2009 1:28:54AM PDT | 11-01-2009 1:28:54AM PDT |
| Base Time Zone (Standard) | 11-02-2009 12:28:54AM PST | 11-02-2009 12:28:54AM PST |
| Base Time Zone (Legal) | 11-02-2009 12:28:54AM PST | 11-02-2009 12:28:54AM PST |
| Referenced Time Zone (Standard) | 11-02-2009 12:28:54AM EST | 11-02-2009 12:28:54AM EST |
| Referenced Time Zone (Legal) | 11-02-2009 12:28:54AM EST | 11-02-2009 12:28:54AM EST |
| Stored in Base Time Zone (Standard) - displayed in another Time Zone | 11-02-2009 2:28:54AM CST | 11-02-2009 2:28:54AM CST |
| Stored in Base Time Zone (Legal) - displayed in another Time Zone | 11-02-2009 2:28:54AM CST | 11-02-2009 2:28:54AM CST |
| Stored in a Referenced Time Zone (Standard) - displayed in another Time Zone | 11-01-2009 11:28:54PM CST | 11-01-2009 11:28:54PM CST |
| Stored in a Referenced Time Zone (Legal) - displayed in another Time Zone | 11-01-2009 11:28:54PM CST | 11-01-2009 11:28:54PM CST |

When an element is associated with a time zone the corresponding time zone name is displayed along with the value. This is specifically useful in the example referenced as "Base Time Zone (Standard) - Duplicate hour". The time 1:28 is ambiguous without a time zone label. Legally, November 1, 2009 1:28 AM occurs twice because daylight savings time (DST) is removed at 2:00 AM. The time zone name indicates whether the first or second hour is specified for the time zone.

### Duration Formatting

**Syntax**

```
oraType="duration"
```

This function is used to display time duration. For an input element, the value entered by the user is translated from minutes to hour and minutes as appropriate. For example, an entered value of '90', is converted to '00:01:30' when tabbing out of the input field.

```
<html>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
      <td>Duration: </td>
      <td><span oraField="duration" oraType="duration"></span></td>
   </tr>
   <tr>
      <td>Duration: </td>
      <td><input oraField="duration" oraType="duration"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <duration>90</duration>
</root>
</xml>
</html>
```

HTML rendered.

Duration: 00:01:30

Duration: 00:01:30


Copyright © 2026, Oracle and/or its affiliates.

### Day in Month Formatting

| Syntax |
| --- |
| `oraType="dayInMonth"` |

This function is used to display a concatenated day and month.

```
<html>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
      <td>Day In Month: </td>
      <td><span oraField="dayMonth" oraType="dayInMonth"></span></td>
   </tr>
   <tr>
      <td>Day In Month: </td>
      <td><input oraField="dayMonth" oraType="dayInMonth"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <dayMonth>0228</dayMonth>
</root>
</xml>
</html>
```

HTML rendered.

Day In Month: 02-28

Day In Month: 02-28

### Month In Year Formatting

| Syntax |
| --- |
| `oraType="monthInYear"` |

This function is used to display a concatenated month and year.

```
<html>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
      <td>Month In Year: </td>
      <td><span oraField="month" oraType="monthInYear"></span></td>
   </tr>
   <tr>
      <td>Month In Year: </td>
      <td><input oraField="month" oraType="monthInYear"/></td>
   </tr>
</table>
</body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<xml>
<root>
   <month>200811</month>
</root>
</xml>
</html>
```

HTML rendered.

### Monetary Formatting

This function is used to display a number as a monetary amount. See the table for configuration options with respect to the currency. For input elements, an error is issued if a non-numeric value is entered.

| Syntax | Description |
|---|---|
| `oraType="money: "` | Directly specify a currency code after the colon. |
| `oraType="money;currencyRef: "` | Reference an XPath (after the colon) for an element that references a currency code. |
| `oraType="money"` | If no currency or currency reference is specified, the installation currency is used. |

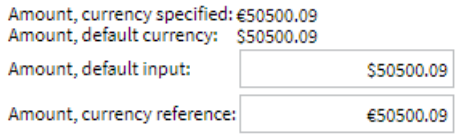
> **Note**
>
> You must specify the stylesheet reference cisDisabled in the map's header for right alignment. The stylesheet controls how the field will be rendered. If you want to alter the rendering you must override the oraMoney style.

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Amount, currency specified:</td>
      <td><span oraType="money:EUR" oraField="totalAmt"></span></td>
   </tr>
   <tr>
      <td>Amount, default currency:</td>
      <td><span oraType="money" oraField="totalAmt"></span></td>
   </tr>
   <tr>
      <td>Amount, default input:</td>
      <td><input oraType="money" oraField="totalAmt"/></td>
   </tr>
   <tr>
      <td>Amount, currency reference:</td>
      <td><input oraType="money;currencyRef:cur" oraField="totalAmt"/></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
</table>
</body>
<xml>
<root>
   <totalAmt>50500.09</totalAmt>
   <cur>EUR</cur>
</root>
</xml>
</html>
```

HTML rendered




### Number Formatting

This function is used to display a number or validate an input value. For input elements, the system will return an error if a non-numeric value is entered.

| Syntax |
| --- |
| `oraType="number"` |

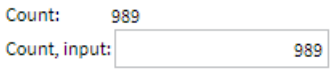
> **Note**
>
> You must specify the stylesheet reference cisDisabled in the map's header for right alignment. The stylesheet controls how the field will be rendered. If you want to alter the rendering you must override the oraNumber style.

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Count:</td>
      <td><span oraType="number" oraField="count"></span></td>
   </tr>
   <tr>
      <td>Count, input:</td>
      <td><input oraType="number" oraField="count"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <count>989</count>
</root>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xml>
</html>
```

HTML rendered.




### FK Reference Formatting

By default, when an element with an **fkRef** oraType is displayed, an info string, context menu, navigation, and search are enabled (if the FK reference has been configured accordingly). Syntax is provided to allow you to selectively turn off any of these features.

Note that you can enable the foreign key hyperlink separately as well, refer to Embed Framework Navigation for more information. The various attributes used to control foreign key reference functionality are as follows. Note that in every case, the default value is **true**. A value of **false** should be used to disable the feature.

| Syntax |
| --- |
| `oraType="fkRef:true\|false; info:true\|false; context:true\|false; navigation:true\|false; search:true\|false"` |

- **fkRef**. A value of 'true' enables all of the following foreign key reference processing. Use a value of 'false' to disable automatic FK Reference processing.
- **info**. A value of 'true' renders an information string on the UI map, if applicable.
- **context**. A value of 'true' renders a context menu to appear before the foreign key reference element, if applicable.
- **navigation**. A value of 'true' causes the information string to be rendered as a hyperlink, if applicable. Clicking the hyperlink navigates to the appropriate page.
- **search**. A value of 'true' displays a search icon that launches the search zone if applicable.

> **Note**
>
> Foreign key navigation and context menu functionality is only available for UI maps presented in a portal zone. UI Maps presented during BPA script processing cannot support navigation options. Search functionality is only available for input HTML elements.

- UI Map schema:

```
<schema>
  <bo fkRef="F1-BOMO"/>
</schema>
```

- UI Map HTML:

```
<html>
<body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Business Object</td>
      <td><span oraField="bo" oraType="fkRef:true; info:true;
context:true; navigation:true;"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <bo>F1-COUNTRY</bo>
</root>
</xml>
</html>
```

- HTML rendered.

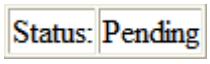
### Lookup Formatting

This function is used to display the description of a lookup value.

| Syntax | Valid Values |
|---|---|
| `oraType="lookup: "` | Lookup field name after the colon. |

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Status:</td>
      <td><span oraField="status" oraType="lookup:BATCH_JOB_STAT_FLG"></
span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <status>PD</status>
</root>
</xml>
</html>
```

HTML rendered.




### Extendable Lookup Formatting

This function is used to display the description of an extendable lookup value.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Valid Values |
| --- | --- |
| `oraType="lookupBO: "` | Business Object code name after the colon. |

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Value:</td>
      <td><span oraField="status" oraType="lookupBO:F1-DeviceDisplayTypes"></
span></td>
   </tr>
</table>
</body>
<xml>
<root>
    <status>oraTablet</status>
</root>
</xml>
</html>
```

HTML rendered.

| Value: | Tablet size |
| --- | --- |

**Characteristic Type Formatting**

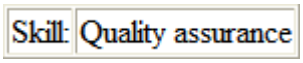
This function is used to display the description of a predefined characteristic value.

| Syntax | Valid Values |
| --- | --- |
| `oraType="charType: "` | Characteristic Type code after the colon. |

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td>Skill:</td>
        <td><span oraType="charType:CM-SKILL" oraField="skill"></span></td>
    </tr>
</table>
</body>
<xml>
<root>
    <skill>10</skill>
</root>
</xml>
</html>
```

HTML rendered.


Copyright © 2026, Oracle and/or its affiliates.




**Control Table Formatting**

This function is used to display the description of a control table that has an associated language table.

| Syntax | Valid Values |
|---|---|
| `oraType="table: "` | Table code after the colon. |

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Currency:</td>
      <td><span oraType="table:CI_CURRENCY_CD" oraField="curr"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <curr>USD</curr>
</root>
</xml>
</html>
```

HTML rendered.

**Add / Remove Grid Formatting**

| Syntax | Description |
|---|---|
| `oraType="addGridRow"` | The addGridRow function is used to build "insert row" dialog into the UI map.<br>• An 'add' image is displayed.<br>• Clicking the image inserts a new row in the grid.<br>• If the list is empty, by default, an empty grid row is automatically be added. This means that the user will always see at least one grid row when this attribute is used. |
| `oraType="deleteGridRow"` | The deleteGridRow function is used to build "delete row" dialog into the UI map.<br>• A 'delete' image is displayed.<br>• Clicking the image removes the adjacent row from the grid. |

> **Note**
>
> Because add and delete dialogs are relevant only inside a table, these attributes must be specified within a <td> element.


Copyright © 2026, Oracle and/or its affiliates.

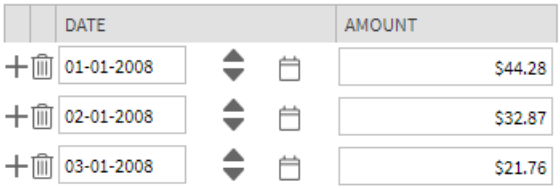
> **Warning**
>
> These attributes are designed to work with the business object action of 'replace' rather than 'update'. Therefore, if the map contains a grid, the business object action of 'replace' must be used to update the business object. Refer to BO Replace Action for more information.

Example:

```
<html>
<head>
<title>Demonstrate Grid Add and Grid Delete OraTypes</title>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table oraList="listEntry">
   <thead>
      <tr>
         <th/>
         <th/>
         <th><span>Date</span></th>
         <th><span>Amount</span></th>
      </tr>
      <tr/>
   </thead>
   <tr>
      <td oraType="addGridRow"></td>
      <td oraType="deleteGridRow"></td>
      <td>
         <input oraField="date" oraType="date"></input>
      </td>
      <td align="right">
         <input oraField="amount" oraType="money"></input>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <listEntry>
      <date>2008-01-01</date>
      <amount>44.28</amount>
   </listEntry>
   <listEntry>
      <date>2008-02-01</date>
      <amount>32.87</amount>
   </listEntry>
   <listEntry>
      <date>2008-03-01</date>
      <amount>21.76</amount>
   </listEntry>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

HTML rendered.




### Unformatted Elements

This function is used to display the contents of an element that contains 'raw' data as defined for the schema element being rendered.

| Syntax |
| --- |
| `oraType="raw"` |

- UI Map schema:

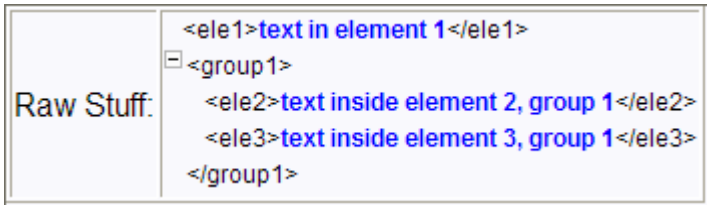
```
<schema>
    <rawStuff type="raw"/>
</schema>
```

- UI Map HTML:

```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>Raw Stuff:</td>
      <td><span oraType="raw" oraField="rawStuff"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <rawStuff>
      <ele1>text in element 1</ele1>
      <group1>
         <ele2>text inside element 2, group 1</ele2>
         <ele3>text inside element 3, group 1</ele3>
      </group1>
   </rawStuff>
</root>
</xml>
</html>
```

HTML rendered.


Copyright © 2026, Oracle and/or its affiliates.




### String Formatting

This function is used to display the contents of an element, as XML pretty-print, when the element contains escaped XML.

| Syntax |
| --- |
| `oraType="xmlString"` |

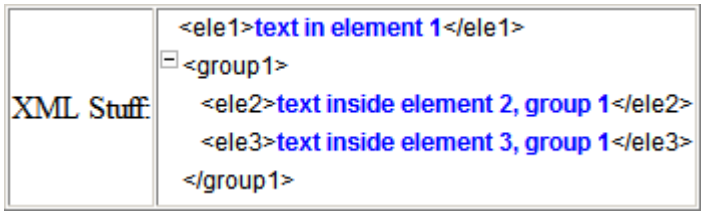
> **Note**
>
> It is not required, but the pretty print of the rendered XML is enhanced if you specify the stylesheet reference cisDisabled in the map's header.

Example:

```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>XML Stuff:</td>
      <td><span oraType="xmlString" oraField="xmlStuff"></span></td>
      </tr>
</table>
</body>
<xml>
<root>
   <xmlStuff>
      <ele1>text in element 1</ele1>
      <group1>
         <ele2>text inside element 2, group 1</ele2>
         <ele3>text inside element 3, group 1</ele3>
      </group1>
   </xmlStuff>
</root>
</xml>
</html>
```

HTML rendered.


Copyright © 2026, Oracle and/or its affiliates.




HTML rendered without oraType="xmlString"

| XML Stuff: | <ele1>text in element 1</ele1><group1><ele2>text inside element 2, group 1</ele2><ele3>text inside element 3, group 1</ele3></group1> |
|---|---|

**HTML Formatting**

This function is used to display the contents of an element as HTML as opposed to plain text. An element defined as oraType="fkref" is automatically rendered as HTML.

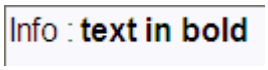
| Syntax |
|---|
| `oraType="html"` |

> **Warning**
>
> To avoid execution of malicious HTML not all HTML tags are supported. The list of supported tags is defined in the "F1-HTMLWhiteList" managed content definition.
>
> If unsupported HTML is detected the entire element is escaped and rendered as plain text. It is therefore recommended to properly escape any source string that contributes to the final HTML element if it is not expected to contain valid HTML. This way only the offending string is escaped and not the entire element.
>
> If the HTML element is composed in scripting refer to the 'escape' function described in the Edit Data Syntax for more information. Use the WebStringUtilities.asHTML java API for escaping text composed in Java.

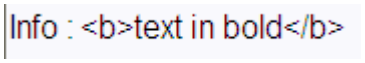
```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table summary="" border="0" cellpadding="1" cellspacing="1">
   <tr>
   <td>Info :</td>
   <td><span oraType="html" oraField="info"></span></td>
   </tr>
</table>
</body>
<xml>
<root>
   <info><b>text in bold</b></info>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

HTML rendered.




HTML rendered without oraType="html"




### Display Labels

#### Derive Label from an Element

This attribute is used to obtain a language sensitive label for a , <td>, or <input> HTML element.

| Syntax | Valid Values |
|---|---|
| `oraLabel=" "` | XPath of an element in the UI map schema. The element must reference either the `mapField=` , `mdField=` , or `label=` attribute. |

> **Note**
>
> You can also define a field directly in your HTML for label definition, refer to Deriving Label from a Field for more information.

> **Note**
>
> If the schema contains multiple attributes, the oraLabel attribute will pick the label to render according to the following hierarchy: The label attribute overrides the mdField attribute, which in turn will override the mapField attribute.

- UI Map schema:

```
<schema>
   <user mapField="USER_ID"/>
   <info type="group" mapXML="CLOB">
      <city label="Metro Area"/>
      <age mdField="AGE_LBL"/>
   </info>
</schema>
```

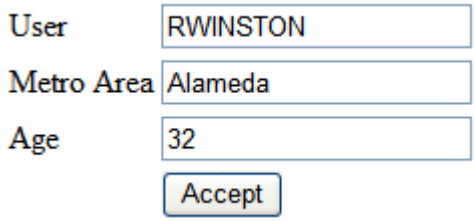
- HTML:

```
<html>
<head><title oraMdLabel="BUS_LBL"></title></head>
<body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<table>
   <tr>
      <td oraLabel="user"></td>
      <td><input oraField="user"/></td>
   </tr>
   <tr>
      <td oraLabel="info/city"></td>
      <td><input oraField="info/city"/></td>
   </tr>
   <tr>
      <td oraLabel="info/age"></td>
      <td><input oraField="info/age"/></td>
   </tr>
   <tr>
      <td/>
      <td><input type="button" oraMdLabel="ACCEPT_LBL"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <user>RWINSTON</user>
   <info>
      <city>Alameda</city>
      <age>32</age>
   </info>
</root>
</xml>
</html>
```

HTML rendered:




### Deriving Label from a Field

This attribute is used to obtain a language sensitive label for a , <td>, <input>, or <title> HTML element. The label text is derived from the field referenced.

| Syntax | Valid Values |
|---|---|
| `oraMdLabel=" "` | MD Field code. |

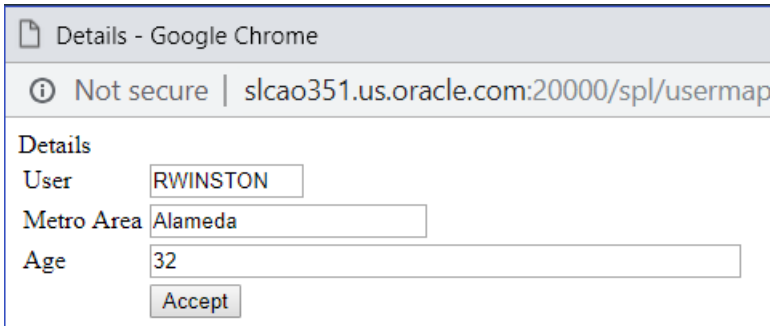
> **Note**
>
> You can also define labels derived from the map's schema definition, refer to Derive Label from an Element for more information.


Copyright © 2026, Oracle and/or its affiliates.

- HTML:

```
<html>
<head><title oraMdLabel="F1_DETAILS_LBL"></title></head>
<body>
<table>
   <tr>
      <td oraLabel="user"></td>
      <td><input oraField="user"/></td>
   </tr>
   <tr>
      <td oraLabel="info/city"></td>
      <td><input oraField="info/city"/></td>
   </tr>
   <tr>
      <td oraLabel="info/age"></td>
      <td><input oraField="info/age"/></td>
   </tr>
   <tr>
      <td/>
      <td><input type="button" oraMdLabel="ACCEPT_LBL"/></td>
   </tr>
</table>
</body>
<xml>
<root>
   <user>RWINSTON</user>
   <info>
      <city>Alameda</city>
      <age>32</age>
   </info>
</root>
</xml>
</html>
```

HTML rendered:




### Enable UI Map Help

The Display Labels section describes ways to derive the label for an element using an underlying MD Field. In addition, if the same MD field contains help text, the system will automatically generate a tool tip adjacent to the element label. Clicking the tool tip allows the user to view the help text.


Copyright © 2026, Oracle and/or its affiliates.

It is possible to change the rendering of the tool tip. Refer to Custom Look And Feel Options for more information

### Search Using a Pop-Up Explorer Zone

### Search Option

This attribute is used to enable search zone functionality for input HTML elements.

| Syntax | Valid Values |
|---|---|
| `oraSearch=" "` | Zone code. |

> **Note**
>
> The oraSearch attribute is similar to the oraType attribute, because it will be 'automatically' included into HTML via the oraSchemaDataTypes attribute. This means that coding the oraSearch attribute into UI Map HTML is only required if a search zone has not been specified in the schema, or in the schema element's FK reference.

- Example of defining the search in the HTML. UI Map's Schema:

```
<schema>
   <uiMap/>
</schema>
```

UI Map's HTML

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap" oraSearch="F1-UISRCH"></td>
   </tr>
</table>
</body>
<xml>
<root>
   <uiMap/>
</root>
</xml>
</html>
```

- Example of defining the search in the schema. UI Map's Schema:

```
<schema>
   <uiMap search="F1-UISRCH"/>
</schema>
```


Copyright © 2026, Oracle and/or its affiliates.

UI Map's HTML

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap"></td>
   </tr>
</table>
</body>
<xml>
<root>
   <uiMap/>
</root>
</xml>
</html>
```

- Example where the FK reference defines the search zone. UI Map's Schema:

```
<schema>
   <uiMap fkRef="F1-UISRC"/>
</schema>
```

UI Map's HTML

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap"></td>
   </tr>
</table>
</body>
<xml>
<root>
   <uiMap/>
</root>
</xml>
</html>
```

In all cases, the HTML rendered is the same.

**Initializing Search Fields**

This optional attribute is used to initialize search zone filters. Multiple filters may be initialized. This attribute can only be used in conjunction with the oraSearch attribute.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Valid Values | Field Value Options | Comments |
|---|---|---|---|
| `oraSearchField=" "` | One or more pairs of field name and field value separated by colon. Each pair is separated by a semi-colon. `oraSearchField="fieldName:fieldValue: ..."` The field name is used to identify the zone filter to initialize when the search is launched. The field name must match the value of the searchField mnemonic specified on a search zone user filter or hidden filter parameter. | No value | If you do not specify a field value, then the value of the input element containing the oraSearchField attribute will be used. |
| | | XPath | Indicate the XPath to the schema element that contains the value to use. |
| | | 'literal' | Indicate a literal value to supply. |

> **Note**
>
> If you do not specify an oraSearchField attribute and the schema element has a search enabled fkRef specified, the framework automatically builds an oraSearchField where the field name is equal to the FK reference's key (MD) fields.

> **Warning**
>
> The pop-up explorer zone can be invoked one of two ways: By clicking on the search button, or by hitting the enter key from the field to the left of the button. If you click on the button then no search field information will be passed to the zone. Search field information is only used to initialize zone filter values when enter is pressed.

Two filter values are initialized as shown in the following example:

```
<schema>
   <bo/>
   <uiMap/>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>UI Map with Search </td>
      <td><input oraField="uiMap" oraSearch="F1-UISRCH"
oraSearchField="MAP_CD; BUS_OBJ_CD:bo;"></td>
    </tr>
</table>
</body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<xml>
<root>
   <bo/>
   <uiMap/>
</root>
</xml>
</html>
```

### Mapping Returned Search Fields

This optional attribute is used to direct values returned by the search zone. Multiple fields may be specified. This attribute can only be used in conjunction with the oraSearch attribute.

| Syntax | Valid Values | Field Value Options | Comments |
|---|---|---|---|
| `oraSearchOut=" "` | One or more pairs of field name and field value separated by colon. Each pair is separated by a semi-colon.<br>`oraSearchOut="field name:xpath target; ..."`<br>The field name is used to identify the search result returned from the query zone. The field name must match the ELEMENT_NAME mnemonic defined within the explorer zone's search results parameter. | No value<br><br>XPath | If you do not specify a field value, then the input element containing the oraSearchField attribute receives the returned value.<br>Indicate the XPath to the schema element that should receive the returned value. |

> **Note**
>
> If you do not specify an oraSearchOut attribute, the framework will build a default where the field name will be set equal to the oraSearchField's field name.

Two values are returned in the following example:

```
<schema>
   <bo/>
   <mo/>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
   <tr>
      <td>BO Search </td>
      <td><input oraField="bo" oraSearch="Z1-BOSRCH"
oraSearchOut="BUS_OBJ_CD; MO_CD:mo;"></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
</table>
</body>
<xml>
<root>
   <bo/>
   <mo/>
</root>
</xml>
</html>
```

### Display Errors

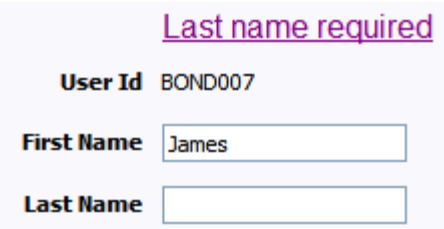
### Display Error Variables

One of the following error variables may be displayed.

| Syntax |
| --- |
| `oraErrorVar="ERRMSG-TEXT"` `oraErrorVar="ERRMSG-LONG"` |
| `oraErrorVar="ERRMSG-CATEGORY"` |
| `oraErrorVar="ERRMSG-NUMBER"` |

```
...
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
...
```

HTML rendered




### Highlight a Field in Error

> **Note**
>
> For more information on throwing an error, refer to the Terminate Statement in the Edit Data Syntax.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Comments |
|---|---|
| `oraError="automate :true\|false; prefix: "` | Specifying `automate:true` automatically enables highlighting of the element in error when issuing an error. Note that **true** is the default and doesn't need to be specified. Specify `automate:false` to turn off field highlighting.<br>The system uses a match on the element name referenced in the error to the element names in the UI map. If the elements in the schema are within an XPath that may not match what is referenced by the error, use `prefix:XPath` to specify that. |

> **Note**
>
> The stylesheet reference cisDisabled must be specified for reference of the oraError style. The stylesheet controls how the field in error will be rendered. If you want to alter the rendering you must override the oraError style.

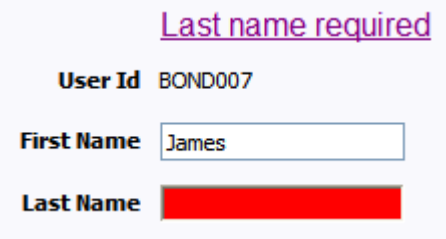
The following HTML example shows that the elements in the map are defined within a group called **boGroup**. The element name returned by the error will not include this group so in order for the field highlighting to work properly the `prefix:` attribute must indicate the group name.

```
<html>
<head>
   <title>User Zone Input</title>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body oraError="automate:true; prefix:boGroup">
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
<table width="100%" border="0" cellpadding="4">
   <tr style="padding-top:30px;">
      <td align="right" class="label">User Id</td>
      <td>
         <span oraField="boGroup/userId" class="normal"/>
      </td>
   </tr>
   <tr>
      <td align="right" class="label">First Name</td>
      <td>
         <input oraField="boGroup/firstName" class="normal"/>
      </td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
   </tr>
   <tr>
      <td align="right" class="label">Last Name</td>
      <td>
         <input oraField="boGroup/lastName" class="normal"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <boGroup>
      <userId>BOND007</userId>
      <firstName>James</firstName>
      <lastName></lastName>
   </boGroup>
</root>
</xml>
</html>
```

HTML rendered, where the error element thrown is equal to 'lastName':




**Overriding the Error Element Name**

In the rare occasion where the element name returned by the error doesn't match the element name in the map, you can add an explicit attribute to indicate the error element name.

| Syntax | Valid Values | Comments |
|---|---|---|
| `oraErrorElement=` | "element name" | The element name referenced here must exactly match the name of the error element assigned when the error was thrown. More than one HTML field can be referenced by the same error element name. |

> **Note**
>
> The stylesheet reference cisDisabled must be specified for reference of the oraError style. The stylesheet controls how the field in error will be rendered. If you want to alter the rendering you must override the oraError style.

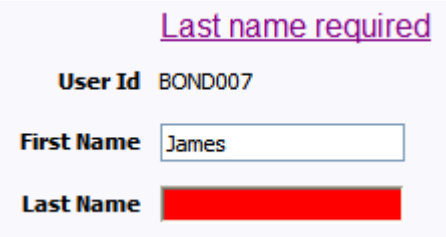
This illustrates a scenario where the element name associated with the error differs from the element name on the map.

```
<html>
<head>
   <title>User Zone Input</title>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</head>
<body>
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
<table width="100%" border="0" cellpadding="4">
   <tr style="padding-top:30px;">
      <td align="right" class="label">User Id</td>
      <td>
         <span oraField="userId" class="normal"/>
      </td>
   </tr>
   <tr>
      <td align="right" class="label">First Name</td>
      <td>
         <input oraField="firstName" class="normal"
oraErrorElement="firstName"/>
      </td>
   </tr>
   <tr>
      <td align="right" class="label">Last Name</td>
      <td>
         <input oraField="familyName" class="normal"
oraErrorElement="lastName"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <userId>BOND007</userId>
   <firstName>James</firstName>
   <familyName></familyName>
</root>
</xml>
</html>
```

HTML rendered.




**Display Error Pop-Up**


Copyright © 2026, Oracle and/or its affiliates.

When the error text is displayed, this function may be used to pop-up the standard error dialog (which displays more information) when a user clicks the error message.

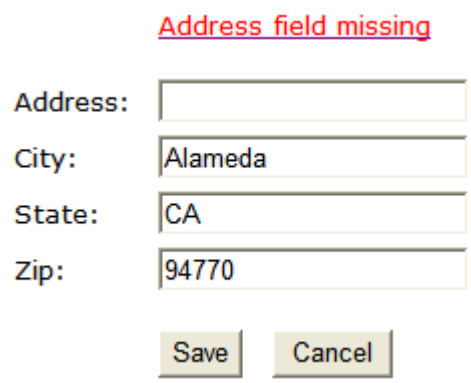
**Syntax**

```
oraShowErrorAlert(); return false;
```

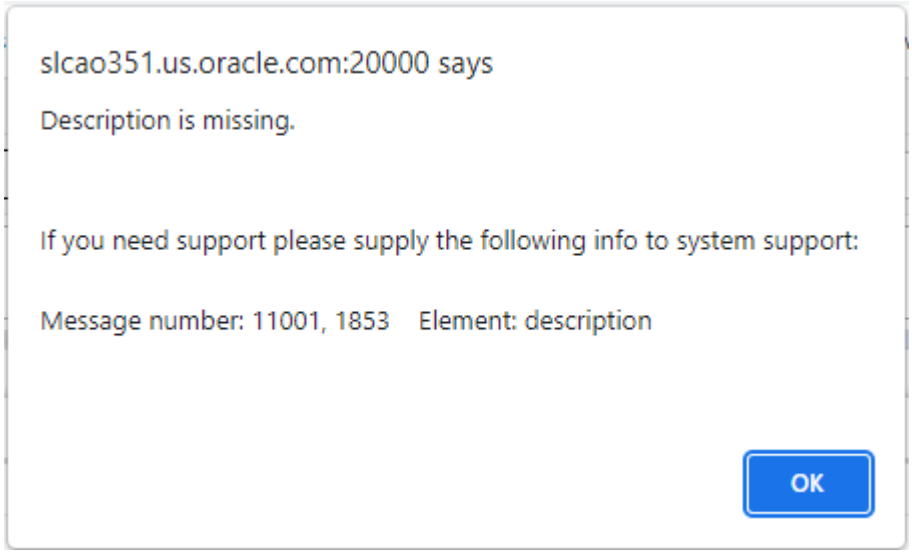
```
<html>
<head>
   <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table width="100%" cellpadding="12">
   <tr class="oraErrorText">
      <td>
         <a href="" onclick="oraShowErrorAlert(); return false;">
            <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span>
         </a>
      </td>
   </tr>
</table>
<table>
   <tr>
      <td >Address:</td>
      <td><input type="text" oraField="address"/></td>
   </tr>
   <tr>
      <td>City:</td>
      <td><input type="text" oraField="city"/></td>
   </tr>
   <tr>
      <td>State:</td>
      <td><input type="text" oraField="state"/></td>
   </tr>
   <tr>
      <td>Zip:</td>
      <td><input type="text" oraField="zip"/></td>
   </tr>
   <tr>
      <td/>
      <td style="padding-top=15px;">
         <oraInclude map="F1-SaveCancelButtons"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <address>123 Main St</address>
   <city>Alameda</city>
   <state>CA</state>
   <zip>94770</zip>
</root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

HTML rendered.




Standard error pop-up dialog launched via click on error message:




**Fire JavaScript for Browser Events**

**Working with the JavaScript Framework**

There are many JavaScript events that can be used within the HTML/Javascript environment. These include events such as onLoad, onBlur, onChange, etc. The UI Map Framework also makes use of some of these events. It is important that any UI Map you develop works with the framework in order to obtain consistent results (events may not always be executed in the same order at all times!).


Copyright © 2026, Oracle and/or its affiliates.

**Warning**

The following describes the recommended approach for safely handling loading and processing field updates in your UI Maps.

If JavaScript is required within an XHTML UI Map or fragment, it will be necessary to bound it within a `![CDATA[ ]]` tag to ensure a valid XML document. Note that the tags themselves may need to be commented out to promote compatibility with older browsers. For example:

```
<script type="text/javascript">
/* <![CDATA[ */

//
// javascript
//

/* ]]> */

</script>
```

**Element Change Event**

| Syntax | Valid Values |
|---|---|
| `oraChange=" "` | A JavaScript function. |

At the time of UI Map load, if there is an event handler already attached to an HTML element, the framework removes it and attaches a combined event handler. The combined handler calls any **framework handler first** and then calls the other (custom) handlers.

**Warning**

Note that the function must not be used to execute logic that will modify the associated field data value again - or an endless loop will occur.

In the following example the **oraInvokeBS** function is executed when the button is clicked.

```
<html>
  <head>
    <title>oraInvokeBS test</title>
  </head>
  <body>
    <table>
      <tr>
        <td>User Id:</td>
        <td>
          <input oraField= "xmlGroup/userId"/>
          <input type="button" value="Search"
oraChange="oraInvokeBS('UserSearch','xmlGroup');"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
          </td>
        </tr>
        <tr>
          <td/>
          <td>Search Results</td>
        </tr>
        <tr>
          <td/>
          <td>
            <table oraList="xmlGroup/searchList">
              <tr>
                <td><span oraField="userId"></span>
                </td>
              </tr>
            </table>
          </td>
        </tr>
      </table>
    </body>
    <xml>
      <root>
        <xmlGroup>
          <userId/>
          <searchList>
              <userId></userId>
          </searchList>
        </xmlGroup>
      </root>
    </xml>
</html>
```

**Page Load Event**

| Syntax | Valid Values |
|---|---|
| `oraLoad=" "` | A JavaScript function. |

**Warning**

When executing oraLoad within a fragment UI map, and you need to execute a JavaScript function during page load (where the function invokes a business object, business service, or service script) you can use the special syntax "`oraLoad[$SEQUENCEID]`". For other special syntax used for map fragments, refer to theConstruct Portal Zone Map Fragments section.

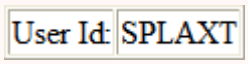
- In the following example, the oraDisplayNone function is executed during page load:

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td oraLoad="oraDisplayNone(item,'userId','')">User Id:
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </td>
            <td><span oraField="userId"></span></td>
        </tr>
    </table>
    </body>
    <xml>
    <root>
        <userId>SPLAXT</userId>
    </root>
    </xml>
    </html>
```

- HTML rendered




**After Page Load Event**

| Syntax | Valid Values |
|---|---|
| `oraAfterLoad=""` | A JavaScript function. |

In the following example the oraGetValueFromScript function is executed after page load.

```
<div>
  <label for="boGroup_parameters_externalSystem" oraLabel="boGroup/
parameters/externalSystem">
  </label>
    <span>
      <select oraSelect="bs:F1-RetrieveExternalSystems"
class="oraInput"
        id="boGroup_parameters_externalSystem" oraField="boGroup/
parameters/externalSystem"
        oraSelectOut="valuePath:results/externalSystem;
descPath:results/description"
        oraSelectIn="outboundMsgType:boGroup/parameters/
outboundMsgType"
        oraAfterLoad

="oraGetValueFromScript(document.getElementById('boGroup_parameters_ext
ernalSystem');">
      </select>
    </span>

</div>
```

### Hide Elements

### Hide Using a Function

The system provides multiple techniques for hiding an HTML element.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Valid Values | Comments |
|---|---|---|
| `oraDisplayNone(item );` | (item, 'XPath', 'value', 'operator') | Used to hide an element based on the value of another element (referenced using its XPath). Enter a value of ' ' to interrogate a blank value. By default the operator is `'='`. This may be set instead to another operator such as `'!='`, `'>'`, or `'<'`. |
| | (item, function name,**true** \| **false**) | Used to indicate a JavaScript function, which must return a Boolean. |
| `oraIsSuppress="true"` | | Use this option for fields that need to be hidden always and not dependent on something else. Set this attribute on the HTML element (div element). |

- Example where the User Id label is hidden when no User Id value exists.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td oraLoad="oraDisplayNone(item,'userId','')">User Id: </td>
        <td><span oraField="userId"></span></td>
    </tr>
</table>
</body>
<xml>
<root>
    <userId></userId>
</root>
</xml>
</html>
```

- Example where the Save button is hidden when the user does not have security access to the action of change ('C') for the application service 'F1-DFLTS'.

```
<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
   <td oraLoad="oraDisplayNone(item, oraHasSecurity('F1-DFLTS', 'C'),
false );">
      <input name="Save" type="button" onclick="oraInvokeBO('CM-
IndividualTaxpayer', null, 'replace')"/>
   </td>
    </tr>
</table>
</body>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<xml>
<root>
    <userId></userId>
</root>
</xml>
</html>
```

### Check User's Security Access

The system provides two functions to check a user's security access to a given application service and access mode. These are commonly used for hiding elements.

| Syntax | Parameters |
|---|---|
| `oraHasSecurity( )` | 'Application Service code'<br>'Access Mode' |
| `oraHasSecurityPath('x','y')` | 'Application Service XPath'<br>'Access Mode XPath' |

See the previous section for an example of the `oraHasSecurity` function. The following shows an example where the status button is hidden when the user does not have security access to the access mode 'ACT' of the application service 'FORMTST'.

```
<html>
<body>
<table>
    <tr>
   <td oraLoad="oraDisplayNone(item, oraHasSecurityPath('appService',
'accessMode'), false );">
      <input oraField="statusLabel" type="button"
onclick="oraRunScript('UpdateState','status');"/>
   </td>
    </tr>
</table>
</body>
<xml>
<root>
    <status>ACTIVATE</status>
<statusLabel>Activate</statusLabel>
<appService>FORMTST</appService>
    <accessMode>ACT</accessMode>
</root>
</xml>
</html>
```

### Invoke Schema Based Services

The system provides functions for invoking a business object, business service or service script.

### Invoke BO Function

This function is used to perform a BO interaction directly from the UI map's HTML. It returns a 'true' or a 'false' depending on whether the invocation encounters an error.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraInvokeBO( )` | 'BO Name' | |
| | 'XPath' or **null** | Identifies a group element via XPath. If you specify the word **null**, then the entire embedded XML object will be passed. |
| | 'action' | Indicate the action to use. Valid values are **add**, **delete**, **read**, **update**, **replace**, **fastAdd** and **fastUpdate**. |

**NOT_SUPPORTED**

Refer to BO Actions for more information about the various BO actions.

Example with the statement invoked in a JavaScript function.

```
function invokeBO {
     if (!oraInvokeBO('F1-User','xmlGroup','read')) {
          oraShowErrorAlert();
          return;
}
}
```

Example with the statement invoked within onClick.

```
<input type="button"  class="oraButton" oraMdLabel="DISPLAY_LBL"
onClick="oraInvokeBO('F1-User');"/>
```

### Invoke BS Function

This function is used to perform a business service interaction directly from the UI map's HTML. It returns a 'true' or a 'false' depending on whether the invocation encounters an error.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraInvokeBS( )` | 'BS Name' | |
| | 'XPath' or **null** | Identifies a group element via XPath. If you specify the word **null**, then the entire embedded XML object will be passed. |

Example with the statement coded within a JavaScript function.

```
function invokeBS {
     if (!oraInvokeBS('F1-UserSearch','xmlGroup')) {
          oraShowErrorAlert();
          return;
  }
}
```


Copyright © 2026, Oracle and/or its affiliates.

Example with the statement invoked via onClick.

```
<input type="button"  class="oraButton" oraMdLabel="DISPLAY_LBL"
onClick="oraInvokeBS('F1-RetrieveInfo');"/>
```

### Invoke SS Function

This function is used to perform a service script interaction directly from the UI map's HTML. It returns a 'true' or a 'false' depending on whether the invocation encounters an error.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraInvokeSS( )` | 'Service Script Name'<br>'XPath' or **null** | Identifies a group element via XPath. If you specify the word **null**, then the document belonging to the parent node will be passed. If the parent node is not enough, then the entire document can always be passed using the following syntax:<br><br>`oraInvokeSS('service script', null, null, [$SEQUENCEID])` |

Example with the statement invoked within a JavaScript function:

```
function invokeSS {
     if (!oraInvokeSS('F1-GetUsers','xmlGroup')) {
          oraShowErrorAlert();
          return;
}
}
```

Example with the statement invoked within onClick.

```
<input type="button"  class="oraButton" oraMdLabel="DISPLAY_LBL"
onClick="oraInvokeSS('F1-GetUserInfo');"/>
```

### Refresh a Rendered Map or Portal Page

### Refresh Map

This function is used to 'Refresh' only the map zone issuing the command.

| Syntax |
|---|
| `oraRefreshMap;` |

```
...
    <tr>
        <td/>
        <td><input type="button" onclick="oraRefreshMap();"
value="Refresh"/></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </tr>
...
```

### Refresh Page

This function is used to refresh all zones in the portal.

| Syntax |
| --- |
| `oraRefreshPage;` |

```
...
    <tr>
        <td/>
        <td><input type="button" onclick="oraRefreshPage();"
value="Refresh"/></td>
    </tr>
...
```

### Embed Framework Navigation

### Navigate using Navigation Option

This function is used to navigate to another page using the information defined on a navigation option.

| Syntax | Parameters |
| --- | --- |
| `oraNavigate( );` | 'Navigation Option code'<br>'Key XPath' |

> **Warning**
>
> This function is only intended for a UI map defined within a portal zone. It should not be used within a UI map launched by a BPA script.

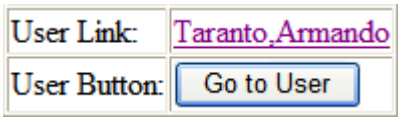
The following example exhibits two possible uses of this function: as a URL and as a button. Note that the UI Map schema must contain a fkRef attribute. Refer to FK Reference Formatting for more information.

```
<schema>
    <userId fkRef="CI_USER"/>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
       <td>User Link: </td>
       <td><a href="" onclick="oraNavigate('userMaint','userId'); return
false;">
         <span oraField="userId" oraType="fkRef:CI_USER"></span></a>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      </td>
    </tr>
    <tr>
        <td>User Button: </td>
        <td><input type="submit" onclick="oraNavigate('userMaint','userId')"
          value="Go to User"/></td>
    </tr>
</table>
</body>
<xml>
<root>
    <userId>SPLAXT</userId>
</root>
</xml>
</html>
```

HTML rendered.




### Launch a View Details Drawer

**Launch a drawer to view additional details for a specific context.**

This function is used for a specific element to launch a side panel Drawer to present related extended information. The launched drawer references a panel name defined by the current portal and that may include one or more zones.

A drawer may be launched from a display zone or a row in a data explorer zone.

> **Warning**
>
> This function is only intended for a UI map defined within a portal zone. It should not be used within a UI map launched by a BPA script. Launching a drawer is also not supported from the Sidebar nor the Assistant portals.

| Syntax | Parameters in UI Map | Parameters in Data Explorer | Comments |
|---|---|---|---|
| Add the following attribute to an HTML element `viewDetails="..."` | `detailPanelName:LITERAL \| x[xpath] \| c[CONTEXTNAME];` | `detailPanelName:LITERAL \| Cx ;` | You must supply a reference to either a panel name defined for the current portal or a zone that is not defined on the portal. |

> **Note**
>
> When referencing a panel name, the


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters in UI Map | Parameters in Data Explorer | Comments |
| --- | --- | --- | --- |
| | `detailZoneCode:LITERAL \| x[xpath] \| c[CONTEXTNAME];` | `detailZoneCode:LITERAL \| Cx ;` | drawer width is explicitly defined on the portal for that panel. When referencing a zone, drawer width is determined by the zone's width. Use a 1/3 width for a small drawer, a 2/3 width for a medium drawer, and a full width for a large drawer.<br><br>From a UI map, the panel name or zone may be specified as a literal without any quotes, as an xpath to an element in the schema or a reference to a global or portal context name.<br>From a data explorer column the non-literal reference may be to another column. |


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters in UI Map | Parameters in Data Explorer | Comments |
|---|---|---|---|
| | `CONTEXTNAME: LITERAL \| x[xpath] \| c[CONTEXTNAME];`<br>The context name can also be a reference to an element `x[xpath]` | `CONTEXTNAME: LITERAL \| Cx;` | Optionally provide values to one or more portal context names that should be set before the drawer is launched.<br>From a UI map, the value may be specified as a literal, an xpath, or a context name.<br>From a data explorer column the non-literal reference may be to another column. |
| | `showPanelIcon :true\|false;` | | By default, including the "view details" attribute on an HTML element also renders a designated icon for launching the drawer. When the launching of the drawer should be made from the HTML element itself, use this parameter to request that the icon is not shown. For example, you can turn the icon off when launching the drawer should be from a "View more information" link element. |

This attribute can be added to most standard HTML elements ( , , etc.) and the new icon, if not disabled, will be presented after the specified element. The following examples shows how to launch a drawer with and without the standard icon and how to provide values of various types.

```
<span oraField="user" id="user" viewDetails="detailPanelName:VIEWDET;
USERID:x[user];"/>
<span oraMdField="MORE_DETAIL" viewDetails="detailPanelName:x[panelName];
showPanelIcon:false; UOM:c[UOM];"/>
<span oraField="user" id="user" viewDetails="detailZoneCode:DETLSZN;
USERID:x[user];"/>
```

### Launch a drawer via a function

The system also provides a function **oraViewDetails** that may be used to launch a drawer. This supports cases where the portal name or the context name needs to be derived. It also supports the case where an object may not have a drawer defined. It may be that if a drawer is defined then launch it, otherwise, navigate to a related page. The syntax is the same as the viewDetails syntax for a UI map. Refer to the UI map F1-LaunchDrawer for an example.

### Launch BPA Script

| Syntax | Parameters | Comments |
|---|---|---|
| `oraRunScript( );` | 'BPA script code'. | |
| | 'XPath Element' | One or more element values may be passed into the BPA where it may be referenced as temporary variables. |


Copyright © 2026, Oracle and/or its affiliates.

> **Warning**
>
> This function is only applicable to UI maps displayed in portal zones. UI maps launched within a running BPA script cannot directly launch another BPA script from the UI map's HTML. Instead, return a value from the UI map and execute a Perform Script or Transfer Control step type.

> **Note**
>
> It is incumbent on the script author to pull information out of temporary storage in the initial steps of the script.

In the following example, a temporary variable named 'personId' is created with value '1234567890' and the BPA script named 'Edit Address' is launched.

```
<html>
<body>
<table>
   <tr>
      <td>
         <div oraField="address"></div>
         <span oraField="city"></span>
         <span>,</span>
         <span oraField="state"></span>
         <span oraField="zip"></span>
         <span oraField="country"></span>
         <a href="" onClick="oraRunScript('Edit
Address','personId');">edit</a>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <personId>1234567890</personId>
   <address>123 Main St</address>
   <city>Alameda</city>
   <state>CA</state>
   <zip>94770</zip>
</root>
</xml>
</html>
```

HTML rendered.

123 Main St
Alameda, CA 94770 edit

**Launch BPA Script With Values**

This function is used to launch a BPA, providing name/value pairs to push into temporary storage. Multiple values can be passed. The BPA script can then reference the temporary variables by name.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraRunScriptWithValues( );` | 'BPA script code'.<br>'XPath Element Name':value | <br>One or more pairs of element names and values. |

> **Note**
>
> You would use this JavaScript function, instead of oraRunScript, if you need to push values to the BPA script that are not located in the UI Map's XML structure.

In the example below, a JavaScript function named 'editUser()' is responsible for launching the BPA script named 'UserEdit'. The temporary variable named 'userId' will be created with value 'CMURRAY'.

```
<html>
<head>
<script type="text/javascript">

function editUser() {
    var values = {'userId': 'CMURRAY'};
    oraRunScriptWithValues('UserEdit', values);
    return;
}

</script>
</head>
<body>
...
</body>
</html>
```

### Exit UI Map with Bound Values

This function is used to exit a UI Map. When you quit the map you can specify a value to return to the script and, in addition, whether to return updated XML.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraSubmitMap( );` | 'Return Value'<br>Boolean value | <br>Indicates if the updated XML should be returned. Default is **true**. |

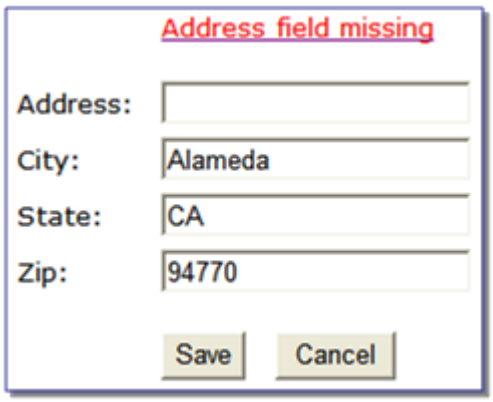
In the following example, the Save button will return updated information, the Cancel button will not.

```
<html>
<body>
<table>
   <tr>
      <td/>
      <td style="padding-bottom:15px;">
         <a href="" onclick="oraShowErrorAlert(); return false;">
```


Copyright © 2026, Oracle and/or its affiliates.

```
         <span oraErrorVar="ERRMSG-TEXT"></span></a>
      </td>
   </tr>
   <tr>
      <td >Address:</td>
      <td><input type="text" oraField="address"/></td>
   </tr>
   <tr>
      <td>City:</td>
      <td><input type="text" oraField="city"/></td>
   </tr>
   <tr>
      <td>State:</td>
      <td><input type="text" oraField="state"/></td>
   </tr>
   <tr>
      <td>Zip:</td>
      <td><input type="text" oraField="zip"/></td>
   </tr>
   <tr>
      <td/>
      <td style="padding-top=15px;">
         <input type="button" value="Save" onClick="oraSubmitMap('SAVE');"/>
         <input type="button" value="Cancel"
onClick="oraSubmitMap('CANCEL',false);"/>
      </td>
   </tr>
</table>
</body>
<xml>
<root>
   <address>123 Main St</address>
   <city>Alameda</city>
   <state>CA</state>
   <zip>94770</zip>
</root>
</xml>
</html>
```

Save and Cancel buttons rendered:





Copyright © 2026, Oracle and/or its affiliates.

### Include a Map Fragment

This function is used to embed a map fragment within another UI map. Note that it is possible to use the include node within a map or a map fragment.

| Syntax | Parameters | Comments |
|---|---|---|
| `<oraInclude map=' ' prefixPath=' '/>` | map='UI Map Code'<br>prefixPath='Xpath' | Optionally specify an xpath prefix to be appended onto every included oraField, oraLabel, oraList, oraSelect valuePath and descPath, oraDownloadData, and oraUploadData attribute value defined within the included UI map fragment's HTML. |

> **Note**
>
> This functionality only applies to XPath attribute values when those values do not appear beneath an oraList attribute. Any XPath value within a table containing an oraList attribute will not be affected by a prefixPath.

- An example of a map fragment with two buttons, named 'F1-SaveCancelButtons'.

```
<input onClick ="oraSubmitMap('SAVE');" oraMdLabel="SAVE_BTN_LBL"
class="oraButton"
type="button"/>
<input onClick ="oraSubmitMap('CANCEL',false);" oraMdLabel="CANCEL_LBL"
class="oraButton"
type="button"/>
```

- An example of a map that includes the map fragment named 'F1-SaveCancelButtons'.

```
...
<tr>
   <td colspan="2" align="center">
<oraInclude map="F1-SaveCancelButtons"/>
   </td>
</tr>
...
```

### Include a Web Component

A web component is a custom HTML element that produces a specific user interface feature. Each supported web component typically has one or more defined HTML attributes that affect how the web component functions and how it appears. The more complex web components may have a corresponding system configuration object that defines the behavior and information displayed by the resulting user interface element. For example the web component


Copyright © 2026, Oracle and/or its affiliates.

that produces a "tree" of information has a Tree configuration table, which defines the details about how to retrieve information for the tree and how to display each node.

**Common Attributes**

Some common attributes that may be used for each web component are:

- A reference to the specific instance of the configuration object that governs the details to produce.
- The ability to pass in context data as name and value pairs.
- A 'mode' that support showing a preview of the feature

Some attributes support a **single value** that is referenced. For these types of attributes, the value may be provided using one of the following references:

- Literal value. The value can be referenced directly. Example:

  ```
  treename="F1-Tree"
  ```

- XPath Reference. XPath references must use the explicit syntax `x[value]`. Example:

  ```
  treename="x[migrationTree]"
  ```

- Global or Portal context reference. These references must use the explicit syntax `c[value]`.

  ```
  treename="c[TREE_NAME]"
  ```

- Field label reference. These references must use the explicit syntax `md[value]`.

  ```
  text=md[F1_ADD_LBL]"
  ```

- Field help text reference. These references must use the explicit syntax `mdh[value]`.

  ```
  text=mdh[F1_OPERATIONS_LBL]"
  ```

For **complex attributes** that support one or more name / value pairs, the syntax for defining the names and values follows the same rules as above for literal values, XPath references and context. Colons (":") should be used to separate the name from the value and semi-colons (";") separate each pair.

```
context="FIELD_NAME1:x[value1];FIELD_NAME2:F1AB;x[fieldName3]:c[FIELD_NAME3];"
```

A given web component may have additional HTML attributes specific to its functionality.

> **Note**
>
> Any UI map that defines a web component must include the F1-OJETLIBSR map fragment within the "body" tag.

```
<body>
<oraInclude map="F1-OJETLIBSR"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
..
</body>
```

> **Note**
>
> Standard HTML attributes may also be provided (e.g ID=".." and style=".."). These are defined as normal. It should be noted that the web components themselves may control some styling aspects directly and thus override any style settings provided on the HTML web component. For example, the Contextual Insights configuration provides settings for text color that would override any color defined in the HTML.

The following sections describe each supported web component.

### Trees

Refer to Trees for more information about tree configuration.

| Syntax | Parameters | Comments |
|---|---|---|
| `<ou-tree...></ou-tree>` | `treename="singleValue"` | You must supply a reference to the name of the tree object that defines the configuration of the tree to display. This attribute expects a single value that follows the syntax described in the single value section above. |
| | `context="..."` | Optionally pass in one or more name / value pairs. The syntax follows the example shown in the complex attributes information above. |

The following example uses an explicit reference to a tree and does not pass any context. The particular tree relies on global context, which the tree node algorithm has access to.

```
<ou-tree treename="C1-PremiseTree"></ou-tree>
```

### Contextual Insight

Refer to Contextual Insights for information about the configuration related to defining contextual insights.

| Syntax | Parameters | Comments |
|---|---|---|
| `<ou-insights ...></ou-insights>` | `insightType="singleValue"` | Use this attribute to specify an insight type to display. Either this attribute or the insight class attribute must be supplied. This attribute expects a single value that follows the syntax described in the single value section above. |
| | `insightClass="singleValue"` | Use an insight class to display all the insight types for the insight group related to this class. This attribute expects a single value that follows the syntax described in the single value section above. |
| | `context="..."` | Optionally pass in one or more name / value pairs. The syntax follows the example shown in the complex attributes information above. |


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
| --- | --- | --- |
| | `mode="preview"` | Optionally pass in a mode of **preview** to indicate to the underlying insight algorithm that prepares the data that the insight is being called in preview mode, with no context available. Ideally the algorithm returns sample information. |
| | `class=" "` | Optionally provide a CSS helper class to control aspects of the rendered insight. Supported values: `class="card-full-width"` - Applicable for card insights. Use this option class to ensure that the width for a card insight adjusts to the width of the zone that contains it. Card insights that don't set this parameter use a fixed width. |

The following example is from the insight preview map where the insight type is the one in context. No context data is needed and the mode is 'preview'.

```
<ou-insights insightType="x[insightType]" mode="preview"></ou-insights>
```

The following example uses an insight class and passes in context values from XPath elements with explicitly defined context names.

```
<ou-insights insightClass="C1CI"
context="PER_ID:x[personId];ACCT_ID:x[accountId];PREM_ID:x[premiseId];"></ou-
insights>
```

### Visualization Widgets

This web component may be used to render various types of visualization widgets such as cards and badges.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
|---|---|---|
| `<ou-visualization .. .></ou-visualization>` | `type="LITERAL \| x[xpath]"` | You must specify the type of widget to render.<br>The valid literal values are:<br>• **`badge`** - a non-clickable badge widget<br>• **`card`** - A card provides a quick view on an issue or a task. At a minimum, it includes a primary text describing the highlighted topic. It may also include optional components: secondary and tertiary lines of text, a badge, a over-line text adjacent to the badge, an icon and action buttons.<br>• **`message`**- a message text along with an optional icon.<br>The type may be specified by a reference to an element that contains one of the valid values. |
| | `color="`<br>`badge:x[xpath]\|LITERAL;`<br>`icon:x[xpath]\|LITERAL;"` | Complex attribute that sets the color option to use for various aspects of the widget.<br>Note that the reference is not a color option and not a specific color.<br>The **`badge:`** attribute is applicable to a **badge** and a **card**.<br>The **`icon:`** attribute is applicable to a **card** and a **message**. |
| | `imgRef="x[xpath]\| LITERAL"` | References an icon defined as a Display Icon.<br>Applicable to a **card** and a **message**. |
| | `imgSize="x[xpath]\| LITERAL"` | May be used to override the standard size of the icon.<br>Valid values are: s (small), m (medium) and l (large).<br>Applicable to a **card** and a **message**. |
| | `text="`<br>`badge:x[xpath]\|LITERAL;`<br>`overline:x[xpath]\| LITERAL;`<br>`main:x[xpath]\|LITERAL;`<br>`secondary:x[xpath]\| LITERAL;`<br>`tertiary:x[xpath]\| LITERAL;` | Applicable to all types of visualization widgets.<br>`badge:` is text to use on a badge.<br>`main:` is the main text to use on the widget.<br>`secondary:` and `tertiary:` are optional 2nd and 3rd text lines on a **card**.<br>`overline:` is text next to the badge on a **card**. |


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
|---|---|---|
| | `actions="x[xpath]"` | This is an xpath reference to a list element describing actions that may be available to the user as part of the widget.<br>Applicable to a **card** widget.<br>The list element must be defined as follows:<br><br>`<somelist type="list"><br><includeDA name="F1-ActionDetails"/><br></somelist>` |

The following is an example of a badge.

```
<ou-visualization type="badge" text="badge:x[text];"
color="badge:x[colorOption];"></ou-visualization>
```

The following is an example of a card with only a primary text.

```
<ou-visualization type="card" text="main:x[main];"></ou-visualization>
```

The following is an example of a card with more components.

```
<ou-visualization type="card"
text="badge:x[text];main:x[mainText];secondary:x[secondaryText]"
color="badge:x[badgeColor];icon:x[iconColor]" imgRef="x[icon]"
actions:x[actions]></ou-visualization>
```

The following is an example of a message with an icon.

```
<ou-visualization type="message" text="main:x[mainText];"
color="icon:x[iconColor]" imgRef="x[icon]"></ou-visualization>
```

Note the following when designing your actions and populating the actions list:

- An action may be of the following types:
  - Navigate to page. The 'action key' in the action's list entry is the navigation option to navigate to.
  - Launch a BPA script. The 'action key' in the action's list entry is the script to launch.
  - Launch a view details panel in a drawer. The 'action key' is the panel name defined on the current portal.
  - Launch a Knowledge Article. The 'action key' is the knowledge article name as defined in the Knowledge Management Articles extendable lookup.
- Context details may be provided for any action as needed.
- By default, actions are rendered as buttons, except for the action to launch a knowledge article that is rendered as the knowledge book icon.


Copyright © 2026, Oracle and/or its affiliates.

- An action may optionally be associated with a role:
  - One action may be set with a **primary** role to highlight it as the main action. This is applicable to redwood view mode only.
  - If the card itself is clickable then one of the actions needs to be set with the **card** role. This action is excluded from the list of buttons.

### Knowledge Article

This web component may be used to associate a UI location with a knowledge article.

Refer to Knowledge Articles for information about knowledge article configuration.

| Syntax | Parameters | Comments |
|---|---|---|
| `<ou-km-link ... ></ou-km-link>` | `kmCode="LITERAL"` | Reference the knowledge article code defined in the **Knowledge Management Articles** extendable lookup. |

The following is an example of a knowledge icon added to a UI map.

```
<ou-km-link kmCode="extendableLookupCode"></ou-km-link>
```

### Button

The button web component provides support for the standard set of button formats used within the system.

| Syntax | Parameters | Comments |
|---|---|---|
| `<ou-button...></ou-button>` | `type=”singleValue”` | You must define the type of button to display. This attribute expects a single value that follows the syntax described in the single value section above. The valid type values are:<br>• `image` - image only<br>• `text` - text only<br>• `imageText` - image followed by text<br>• `menu` - text plus a dropdown caret to open the menu<br>• `imageMenu` - image plus a dropdown caret to open the menu<br>If no type attribute is supplied, a ‘broken button’ image will be displayed. |
| | `img=”singleValue”` | Define the path of the image to display. Either this attribute or the image reference attribute must be supplied for a button type that includes an image. This attribute expects a single value that follows the syntax described in the single value section above.<br>Note that for SVG images, if the reference includes only the SVG file name with no path, the path is assumed to be the Framework SVG repository. |


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
|---|---|---|
| | `imgRef=”singleValue”` | Define the display icon reference of the image to be used. Either this attribute or the image attribute must be supplied for a button type that includes an image. This attribute expects a single value that follows the syntax described in the single value section above. |
| | `text="..."` | Define the various text elements needed by the button. The syntax follows the example shown in the complex attributes information above. The valid text element name values are:<br>• `label` - the label text to be used for the button<br>• `tooltip` - the hover text to be used for the button<br>• `aria-label` - the ARIA label for the button |
| | `menu=”singleValue”` | Define the menu to be used with the button. This attribute must be supplied for a button type that includes a menu control. This attribute expects a single value that follows the syntax described in the single value section above. |
| | `disabled=”singleValue”` | This is a standard HTML attribute. A value of "true" or just the presence of the text "disabled" will result in an initially disabled button. |
| | `security="..."` | Optionally define the security settings for the button. The button will be hidden if the user does not have the appropriate security access. The syntax follows the example shown in the complex attributes information above. The valid security element name values are:<br>• `service` - the application service whose security access governs access to the button<br>• `accessMode` - the specific access mode for the service that determines a more granular level of access to the button |


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
|---|---|---|
| | `clickAction="..."` | Optionally define the action to be taken when the button is 'clicked'. The syntax follows the example shown in the complex attributes information above. There are specific 'name' values for the attribute as follows:<br>• `action` - the action to be taken. Valid values are 'BPA' and 'NAVIGATE'<br>• `script` - the script to be performed when the action is 'BPA'<br>• `navopt` - the navigation option to be used when the action is 'NAVIGATE'<br>The attribute may also include other name/value pairs that define information to be passed into the script or used as navigation context fields.<br>In addition to the security attribute settings, the system will check that the user has access to the script or navigation path before performing the action.<br>The standard HTML 'onClick' attribute may still be used. if the HTML includes an 'onClick'attribute as well as the 'clickAction', the 'onClick' attribute will take precedence |

The following is an example of a simple button with text.

```
<ou-button type="text" text="label:md[F1_BOOKMARK_LBL];"></ou-button>
```

The following is an example of an icon button using a display icon.

```
<ou-button type="image" imgRef="F1CMNT"></ou-button>
```

The following is an example of an image menu button.

```
<ou-button type="imageMenu" imgRef="F1STTGS" menu="CI_CONTEXTSCRIPT"
text="tooltip:md[F1INSTY_VIS_LBL];></ou-button>
```

The following is an example of text button that navigates to another page.

```
<ou-button type="text" text="label:md[ALG_CD];tooltip:md[GOTO_ALG]"
clickAction="action:NAVIGATE; navopt:f1algqTabMenu; ALG_CD:x[algorithm];></ou-
button>
```

### Decorator

The decorator web component may be used to render an icon for decoration purposes only. This is not an actionable item.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Parameters | Comments |
|---|---|---|
| <ou-decorator...></ou-decorator> | img=”singleValue” | Refer to ou-button syntax. |
| | imgRef=”singleValue” | Refer to ou-button syntax. |

The following is an example of a decorator icon using a display icon.

```
<ou-decorator imgRef="F1CMNT"></ou-decorator
```

### Show Schema Default on Add

Default values within in the UI map's schema will be displayed on a UI map's input fields if an embedded <action> node has a value of 'ADD' or blank.

| Syntax |
|---|
| <action>ADD</action> |
| <action> </action> |

The schema default for the <description> element will be displayed:

```
<schema>
    <action/>
    <boGroup type="group">
        <key/>
        <description default="enter description here"/>
    </boGroup>
</schema>


<html>
<body>
<table summary="" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td>Description   </td>
        <td><input oraField="boGroup/description"></td>
    </tr>
</table>
</body>
<xml>
<root>
    <action>ADD</action>
    <boGroup>
        <key/>
        <description/>
    </boGroup>
</root>
</xml>
</html>
```

HTML rendered.


Copyright © 2026, Oracle and/or its affiliates.

### Configure a Chart

In addition to integrating charts with zones, the system supports using the same attributes within the UI Map HTML to configure a graphical representation of an XML list. The designer can control the type, size, position, and contents of the chart using these attributes. For details of the attributes and syntax, refer to Data Explorer Graph Configuration.

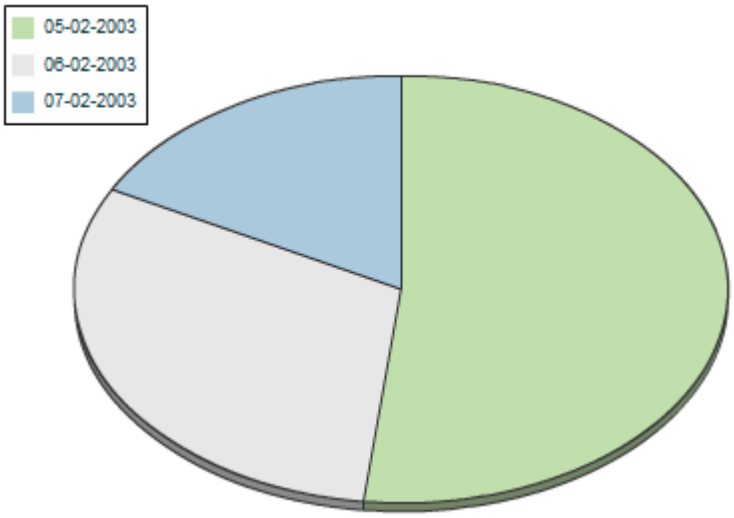
### Graph Examples

- Sample of a pie chart configuration:

```
<html>
<head>
<title>Pie Chart</title>
</head>
<body>

<div style="width:100%; height:290px;"
    oraChart="type:pie;"
    oraChartSeries1="list:set; labelPath:date; amount:amount; "
    oraChartBroadcast="BILL_ID:billId;">
</div>

</body>

<xml>
<root>
  <set>
<date>05-02-2003</date>
<amount>163.24</amount>
<billId>592211649441</billId>
  </set>
  <set>
<date>06-02-2003</date>
<amount>97.29</amount>
<billId>592211649442</billId>
  </set>
  <set>
<date>07-02-2003</date>
<amount>54.38</amount>
<billId>592211649443</billId>
  </set>
</root>
</xml>
</html>
```

- A pie chart rendered for a single series:


Copyright © 2026, Oracle and/or its affiliates.




- Sample of a line, cluster, or stacked graph configuration - each with two series:

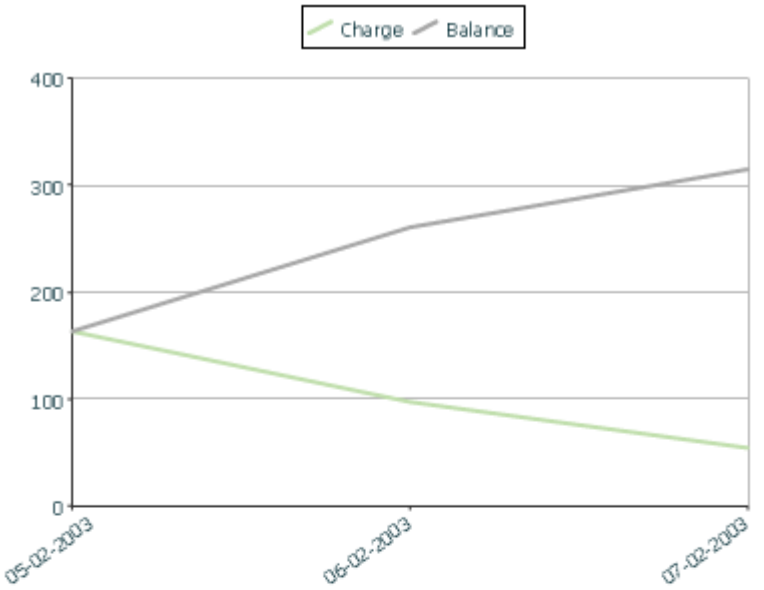
```
<html>
<head>
<title>Stacked Chart</title>
</head>
<body>

<div style="width:100%; height=300px;"
    oraChart="type:line;"
oraChartSeries1="list:set; xaxis:date; label:Charge; amount:amount; "
oraChartSeries2="list:set; xaxis:date; label:Balance; amount:balance; "
    oraChartBroadcast="BILL_ID:billId;">
</div>

<div style="width:100%; height=300px;"
oraChart="type:cluster;"
oraChartSeries1="list:set; xaxis:date; label:Charge; amount:amount; "
oraChartSeries2="list:set; xaxis:date; label:Balance; amount:balance; "
oraChartBroadcast="BILL_ID:billId;">
</div>

<div style="width:100%; height=300px;"
oraChart="type:stacked;"
oraChartSeries1="list:set; xaxis:date; label:Charge; amount:amount; "
oraChartSeries2="list:set; xaxis:date; label:Balance; amount:balance; "
oraChartBroadcast="BILL_ID:billId;">
</div>

</body>

<xml>
<root>
  <set>
<date>05-02-2003</date>
<amount>163.24</amount>
<balance>163.24</balance>
<billId>592211649441</billId>
```

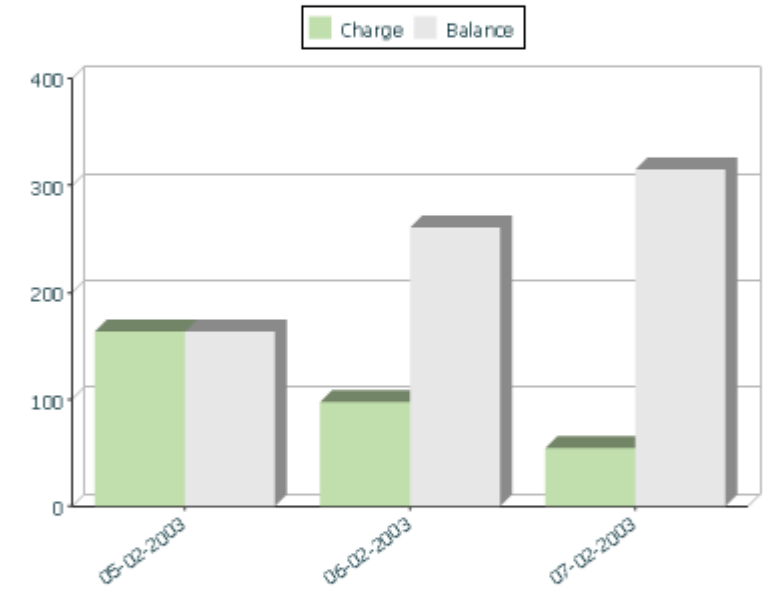

Copyright © 2026, Oracle and/or its affiliates.

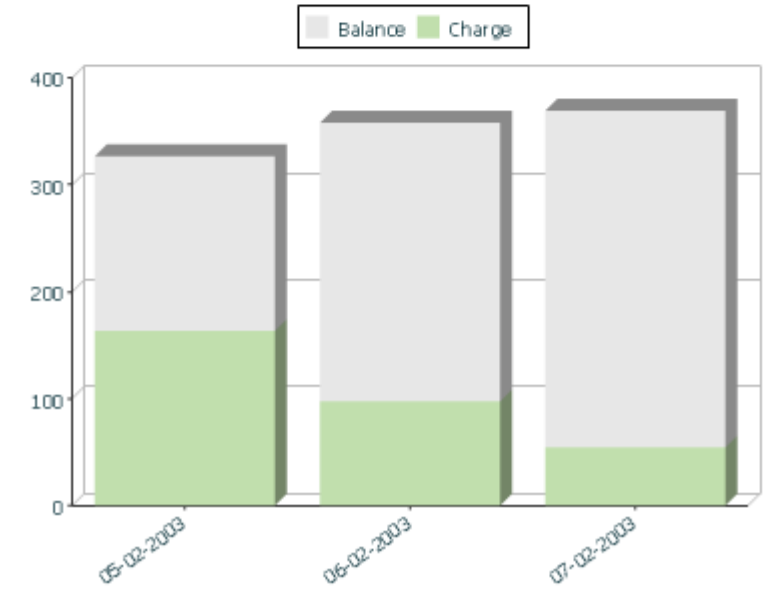
```
  </set>
  <set>
<date>06-02-2003</date>
<amount>97.29</amount>
<balance>260.53</balance>
<billId>592211649442</billId>
  </set>
  <set>
<date>07-02-2003</date>
<amount>54.38</amount>
<balance>314.91</balance>
<billId>592211649443</billId>
  </set>
</root>
</xml>
</html>
```

- Three types of chart rendered for two series each: line, cluster, and stacked.


Copyright © 2026, Oracle and/or its affiliates.





Copyright © 2026, Oracle and/or its affiliates.

### Upload and Download a CSV File

The following HTML attributes can be used to manage both an upload and a download between a list defined within the map's schema and a CSV (comma separated value) file. Note that this technique is only recommended for a small to medium volume of data, for example no more than several hundred rows. For higher volumes, it is recommended to use batch upload / download functionality instead.

The syntax is `oraUploadData="type:embed;path:list xpath;useLabels:true;showCount:true"`

Upload configuration requires you to name a CSV file to be uploaded, and an XML list as target. By convention, each CSV row will create a separate list instance. Each comma-separated field in the file will be uploaded as a separate element in the list. To embed an upload dialog within a map, the **oraUploadData** attribute must be associated with a container element such as a div, td, or span.

The optional `useLabels:true` value indicates that while parsing the upload CSV file, the headers are expected to be labels

> **Note**
>
> If you do not specify the `useLabels:true` value and the XML target element name is "camelCase" then the corresponding spreadsheet header should be title case with a space between words, e.g.;"Camel Case". Letters and special characters are not considered a different word, for example Address1 must be uploaded into the target XML element address1.

Specifying the optional `showCount:true` value will display the number of records uploaded.

> **Caution**
>
> If you are using a grid in conjunction with the **oraUploadData** function, then you must maintain the grid's list with a 'replace' business object action. Refer to BO Replace Action for more information.

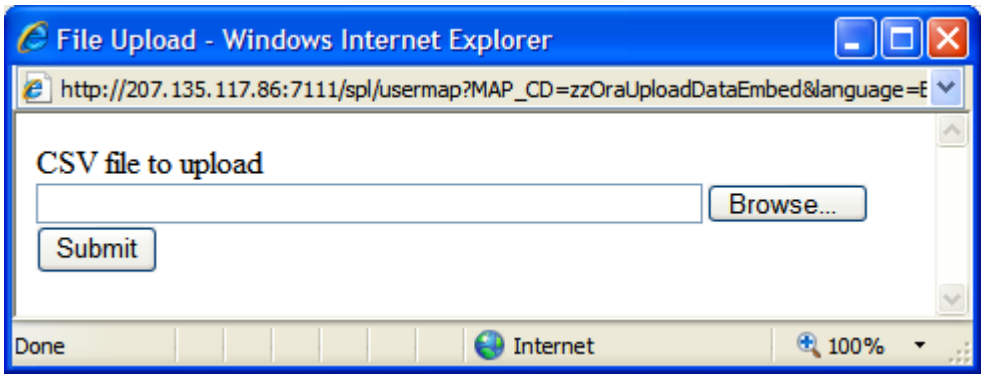
Sample of `oraUploadData="embed"` within a div element.

```
<html>
<head>
    <title>File Upload</title>
</head>
<body>

    <div oraUploadData="type:embed;path:myList"> </div>

</body>

<xml>
<root>
    <myList>
<id>838383930</id>
        <name>Janice Smith</name>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </myList>
    <myList>
<id>737773730</id>
        <name>Bill McCollum</name>
    </myList>
</root>
</xml>
</html>
```

This file upload dialog will be embedded into the body of the page where the oraUploadData is defined.




**oraUploadData="type:popup;path:list xpath;useLabels:true;showOk:true;showCount:true"**

Upload configuration requires you to name a CSV file to be uploaded, and an XML list as target. By convention, each CSV row will create a separate list instance. Each comma-separated field in the file will be uploaded as a separate element in the list. To upload a CSV file using a pop-up dialog, the oraUploadData attribute must be associated with an input element such as a button, text link, or image.

The optional useLabels:true value is used to indicate that while parsing the upload CSV file, the headers are expected to be labels

> **Note**
>
> If you do not specify the useLabels:true value and the XML target element name is "camelCase" then the corresponding spreadsheet header should be title case with a space between words, e.g., "Camel Case". Letters and special characters are not considered a different word, for example Address1 must be uploaded into the target XML element address1.

Specifying the optional showOk:true value will display an "Ok" button once the upload finishes. The popup will stay open until the button is pressed. Additionally, specifying the showCount:true value will display number of records uploaded.

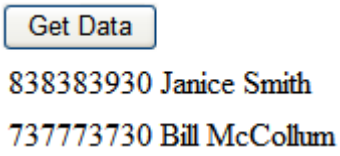
> **Caution**
>
> If you are using a grid in conjunction with the **oraUploadData** function, then you must maintain the grid's list with a 'replace' business object action. Refer to BO Replace Action for more information.


Copyright © 2026, Oracle and/or its affiliates.

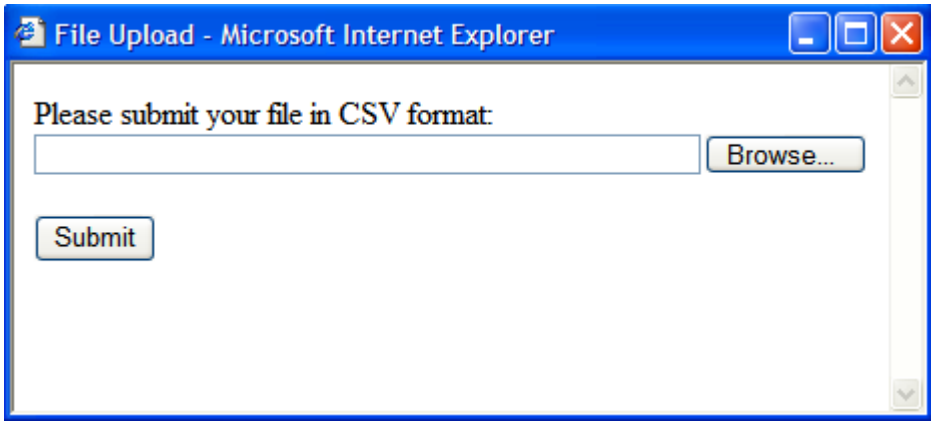
Sample of oraUploadData="popup" associated with a button:

```
<html>
<head>
    <title>File Upload</title>
</head>
<body>
   <input type="button" name="submitButton"
oraUploadData="type:popup;path:myList;" value='Get Data'>
   <table oraList="myList">
      <tr/>
      <tr>
         <td><span oraField="id"/></td>
         <td><span oraField="name"/></td>
      </tr>
   </table>
</body>
<xml>
<root>
   <myList>
      <id>838383930</id>
      <name>Janice Smith</name>
   </myList>
   <myList>
      <id>737773730</id>
      <name>Bill McCollum</name>
   </myList>
</root>
</xml>
</html>
```

HTML Rendered:




Pressing the "Get Data" button will launch a standard file upload dialogue (provided by Framework) as shown below.




**oraDownloadData="list xpath"**


Copyright © 2026, Oracle and/or its affiliates.

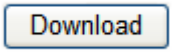
Download configuration requires you to name an XML list to be downloaded. By convention, each list instance will represent a separate row in the created file. By default every element of the list will be comma separated in the file.

> **Note**
>
> The number formatting is based on the user profile setting. For localities where the decimal symbol is a comma, an implementation may configure a property setting (spl.csv.delimiter.useFromDisplayProfile=true) to cause the system to use a semicolon as the delimiter that separates the elements rather than a comma.

Sample of oraDownloadData.

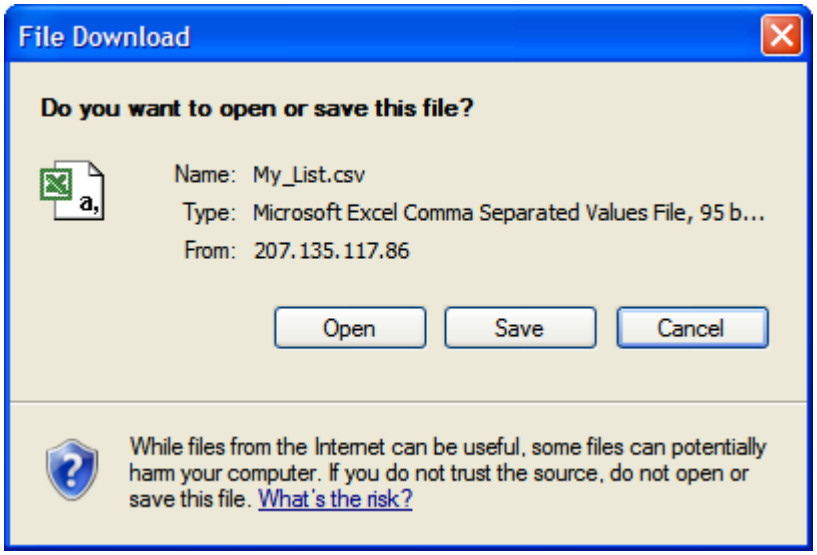
```
<html>
<head>
<title>File Download</title></head>
<body>
<input type="button" name="downloadButton" oraDownloadData="myList"
value="Download"/>
</body>
<xml>
<root>
   <myList>
      <id>881-990987</id>
      <name>John Mayweather</name>
   </myList>
   <myList>
      <id>229-765467</id>
      <name>Anna Mayweather</name>
   </myList>
   <myList>
      <id>943-890432</id>
      <name>Andrew Brewster</name>
   </myList>
</root>
</xml>
</html>
```

HTML Rendered:

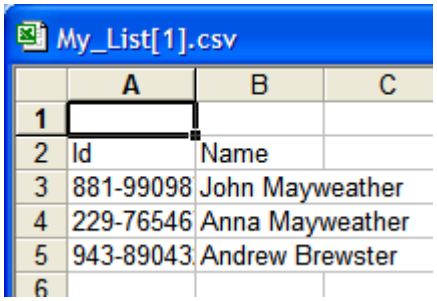



Pressing the "Download" button will launch a standard file download dialogue (provided by Framework) as shown below.


Copyright © 2026, Oracle and/or its affiliates.




A successful download will result in a CSV file:




To download data from a sub list use the attribute oraDownloadDataInList instead of oraDownloadData. The attribute oraDownloadDataInList will have the sub list name. The XPath of the sub list is used to pick data of the specific row from the parent list. Thus only the specific sub list is downloaded.

**oraDownloadDataUseLabels="true"**

The oraDownloadDataUseLabels attribute can be used in conjunction with the oraDownloadData attribute described above. Specify oraDownloadDataUseLabels if you want the generated CSV file to use the element labels for columns headers rather than element names.

### Construct Portal Zone Map Fragments

Portal zones can reference a UI map for the filter area. This UI map is not a complete HTML document, but is instead configured as a UI Map fragment. When constructing a zone map fragment you can reference the following substitution variables. Note that these variables will be dynamically populated at run time with information particular to the map's zone within the portal:

| Variable | Replacement Logic |
|---|---|
| [$ZONEDESCRIPTION] | Zone's description text. |
| [$SEQUENCEID] | Zone's sequence ID. |
| [$ZONENAME] | Zone's name. |
| [$HELPTEXT] | Zone's help text. |


Copyright © 2026, Oracle and/or its affiliates.

| | |
|---|---|
| `[$ZONEPARAMNAME]` | Zone parameter's value (or blank if it has not been specified). |

**Warning**

- These maps make use of the oraInclude tag to incorporate HTML fragments for the framework actions. Refer to the zone type parameters for the UI Map fragments you should include in your HTML.
- If you wish to have the "help text" icon appear next to your zone description, you should have id="title_[$SEQUENCEID]" on the <td> that contains your description.
- If it is necessary to encapsulate JavaScript within a UI Map fragment, it will be necessary to bound the JavaScript within a `![CDATA[ ]]` tag to ensure a valid XML document. Note that the tags themselves may need to be commented out to promote compatibility with older browsers. For example:

```
<script type="text/javascript">

/*<![CDATA[ */

//
//javascript
//
/*]]> */
</script>
```

**Note**

If you wish to preserve the values of a filter input field, within a filter map fragment, for the framework 'Go Back' and 'Go Forward' functionality, you must associate the input field (text box, select, etc.) with a unique HTML id. Input field values associated with a unique id will be captured in the framework's 'memento'. The 'memento' is used to rebuild the input map when the portal zone is navigated to using the 'Go Back' or 'Go Forward' functionality.

**Note**

Many specialized functions exist to manipulate zone behavior, for example:

- `oraGetZoneSequence(zoneName)`. Uses the zone's code to retrieve its sequence number.
- `oraIsZoneCollapsed(sequenceId)`. Uses the zone's sequence to determine if collapsed.
- `oraHandleCollapse(seq)`. Collapse a zone.
- `oraHandleExpand(seq,refresh)`. Expand and/or refresh a zone.

All of these, and many more functions, are located within the JavaScript library userMapSupport.js described below.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When executing oraLoad within a fragment UI map, and you need to execute a JavaScript function during page load (where the function invokes a business object, business service, or service script) you can use the special syntax "oraLoad[$SEQUENCEID]". Refer to the Load Page Event section for more information.

Example of oraLoad[$SEQUENCEID] used within a function:

```
<script type="text/javascript">
function oraLoad[$SEQUENCEID]() {
checkRebateClaimStatus();
}

function checkRebateClaimStatus() {
    var work = id(''analyticsFilterText[$SEQUENCEID]'',
document).cells[0].innerText.split('' '');
    var rebateClaimId = work[work.length - 3];
    id(''rebateClaimId'', document).value = rebateClaimId;
oraInvokeSS(''C1-CheckRCSt'',''checkRebateClaimStatus'', false);
    var statusIndicator = id(''statusInd'', document).value;
    if (statusIndicator == ''C1PE'' || statusIndicator == ''C1ID'') {
       id(''addRebateClaimLine'', document).style.display = '''';
    } else {
id(''addRebateClaimLine'', document).style.display = ''none'';
    }
}
</script>
```

### Invoking a Business Object

The oraInvokeBO function may be used within a zone filter map. It is similar to the command described in Invoke BO Function which allows for a business object to be invoked within the UI map's HTML. Refer to that section for a description of the first three parameters.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraInvokeBO( )` | 'BO Name'<br>'XPath' or **null**<br>'action' | |
| | `null` | This must be specified as the fourth argument. |
| | `[$SEQUENCEID]` | This must be specified as the fifth argument. |
| | `false` | Specify `false`. The 'true' option is no longer applicable. |

Example:

```
oraInvokeBO('CM-User','xmlGroup','read', null, [$SEQUENCEID], false)
```

### Invoking a Business Service


Copyright © 2026, Oracle and/or its affiliates.

The oraInvokeBS function may be used within a zone filter map. It is similar to the command described in Invoke BS Function which allows for a business service to be invoked within the UI map's HTML. Refer to that section for a description of the first two parameters.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraInvokeBS( )` | 'BO Name' | |
| | 'XPath' or **null** | |
| | `null` | This must be specified as the fourth argument. |
| | `[$SEQUENCEID]` | This must be specified as the fifth argument. |
| | `false` | Specify `false`. The 'true' option is no longer applicable. |

Example:

```
oraInvokeBS('CM-UserSearch','xmlGroup', null, [$SEQUENCEID], false)
```

### Invoking a Service Script

The oraInvokeSS function may be used within a zone filter map. It is similar to the command described in Invoke SS Function which allows for a service script to be invoked within the UI map's HTML. Refer to that section for a description of the first two parameters.

| Syntax | Parameters | Comments |
|---|---|---|
| `oraInvokeSS( )` | 'Service Script Name' | |
| | 'XPath' or **null** | |
| | `null` | This must be specified as the fourth argument. |
| | `[$SEQUENCEID]` | This must be specified as the fifth argument. |
| | `false` | Specify `false`. The 'true' option is no longer applicable. |

Example:

```
oraInvokeSS('UserSearch','xmlGroup', null, [$SEQUENCEID], false)
```

### Detecting Unsaved Changes

Use this function to return a Boolean set to true if there are unsaved changes. The system will interrogate the function when the user attempts to navigate and issue a warning accordingly. This function is only needed if a UI map is using custom javascript to manage elements such that the system is not able to detect whether changes have been made. Also note that it's the responsibility of the UI map javascript to manage the values in the Boolean used for this function.

```
function hasUnsavedChanges(){
       return isDirtyFlag;
   }
```

### Hiding Portal Tabs

The product provides the ability to use JavaScript to hide a tab on the current portal based on some condition using the oraAuthorizeTab JavaScript API. This API accepts a function as a parameter and turns off the tab index indicated.


Copyright © 2026, Oracle and/or its affiliates.

For example, the UI Map may have a function to turn off one or more tab indexes.:

```
function overrideTabIndex(index){
    if (index == 2) return false;
    if (index == 3) return false;
  }
```

The JavaScript is referenced “on load”:

```
<body class="oraZoneMap"
onLoad="oraAuthorizeTab(overrideTabIndex);">
```

### Required JavaScript Libraries

All of the functionality described in this document depends on a pair of JavaScript libraries. If you are writing and executing your maps entirely within the UI map rendering framework - you do not need to manually insert the following libraries - the framework will insert them for you when the UI Map is rendered.

> **Warning**
>
> When executing HTML outside of the framework you must include the following references explicitly within your HTML. In addition, the tool you use to render the HTML must have access to a physical copy of privateUserMapSupport.js for bind support.

```
src="privateUserMapSupport.js"
```

Your HTML document must reference this library to execute binding in a stand-alone environment.

> **Warning**
>
> Referencing functions within this JavaScript library is dangerous - because these functions are owned by framework and they may be changed during version upgrade or via the normal patch process.

```
<script type="text/javascript" src="privateUserMapSupport.js"></script>
```

```
src="userMapSupport.js"
```

To take advantage of optional toolset features, you must reference this library.

> **Note**
>
> You can reference the functions within this JavaScript library to write custom functions within the UI map..

```
<script type="text/javascript" src="userMapSupport.js"></script>
```


Copyright © 2026, Oracle and/or its affiliates.

```
onload="oraInitializeUserMap();"
```

To execute binding in a stand-alone environment, you must embed the following onload function into the <body> node.

```
<body onload="oraInitializeUserMap();">
```

# UI Map Standards

### Basic UI Map Templates

All UI Maps share the same basic structure regardless of placement (page area, zone, pop-up) or usage (display only, input).

### Sample XML

All information in this document is based upon the following XML structure.

```
<xml>
    <root>
        <address>123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
</xml>
```

### Display Only UI Map

```
<html>
<head>
    <title oraMdLabel="ADDRESS_LBL"></title>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body class="oraZoneMap">
<table cellspacing="4" width="100%">
    <colgroup>
        <col class="oraLabel oraTableLabel" />
        <col class="oraNormal oraTableData" />
    </colgroup>
    <tr>
        <td oraLabel="address"></td>
        <td oraField="address"></td>
    </tr>
    <tr>
        <td oraLabel="city"></td>
        <td oraField="city"></td>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </tr>
    <tr>
        <td oraLabel="state"></td>
        <td oraField="state"></td>
    </tr>
    <tr>
        <td class="oraSectionEnd" oraLabel="zip"></td>
        <td class="oraSectionEnd" oraField="zip"></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionHeader"
oraMdLabel="CONTACT_LBL"></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionStart oraEmbeddedTable">
            <table oraList="contactInformation" cellspacing="2">
                <thead >
                    <tr>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/type></span>
                        </th>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/number"></span>
                        </th>
                    </tr>
                </thead >
                <tbody>
                    <tr>
                        <td class="oraNormalAlt oraDisplayCell">
                            <span oraField="type"></span>
                        </td>
                        <td class="oraNormal oraDisplayCell">
                            <span oraField="number"></span>
                        </td>
                    </tr>
                </tbody>
            </table>
        </td>
    </tr>
</table>
</body>
<xml>
    <root>
        <address>123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xml>
</html>
```

**Input UI Map**

```
<html>
<head>
    <title oraMdLabel="ADDRESS_LBL"></title>
    <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
</head>
<body>
<table width="100%" cellpadding="12">
  <tr class="oraErrorText">
    <td><a href="" onclick="oraShowErrorAlert(); return false;">
      <span  class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span></a>
    </td>
  </tr>
</table>
<table cellspacing="4" width="100%">
    <colgroup>
        <col class="oraLabel oraTableLabel" />
        <col class="oraNormal oraTableData" />
    </colgroup>
    <tr>
        <td oraLabel="address"></td>
        <td><input type="text" oraField="address"/></td>
    </tr>
    <tr>
        <td oraLabel="city"></td>
        <td><input type="text" oraField="city"/></td>
    </tr>
    <tr>
        <td oraLabel="state"></td>
        <td><input type="text" oraField="state"/></td>
    </tr>
    <tr>
        <td oraLabel="zip"></td>
        <td><input type="text" oraField="zip"/></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionHeader"
oraMdLabel="CONTACT_LBL"></td>
    </tr>
    <tr>
        <td colspan="2" class="oraSectionStart oraEmbeddedTable">
            <table oraList="contactInformation" cellspacing="2">
                <thead >
                    <tr>
                        <th class="oraGridColumnHeaderButton"></th>
                        <th class="oraGridColumnHeaderButton"></th>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/type></span>
                        </th>
                        <th class="oraGridColumnHeader" nowrap="nowrap">
                            <span oraLabel="contactInformation/number"></span>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                        </th>
                    </tr>
                </thead >
                <tbody>
                    <tr>
                        <td oraType="addGridRow"></td>
                        <td oraType="deleteGridRow"></td>
                        <td>
                            <input type="text" oraField="type"/>
                        </td>
                        <td>
                            <input type="text" oraField="number"/>
                        </td>
                    </tr>
                </tbody>
            </table>
        </td>
    </tr>
    <tr>
      <td colspan="2" class="oraSectionStart oraEmbeddedTable">
        <table cellspacing="2">
          <tr>
            <td>
              <input class="oraButton" oraMdLabel="C1_SAVE_LBL" type="button"
                onClick="oraSubmitMap('OK');"/>
            </td>
            <td>
              <input class="oraButton" oraMdLabel="CANCEL_LBL" type="button"
               onClick="oraSubmitMap('CANCEL',false);"/>
            </td>
          </tr>
        </table>
      </td>
    </tr>
</table>
</body>
<xml>
    <root>
        <address>123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
</xml>
</html>
```


Copyright © 2026, Oracle and/or its affiliates.

### Basic HTML and Styles

The basic templates introduced the standard HTML and styles used for UI Maps. These standards are described individually in the following sections.

### Stylesheets

The styles to apply the standard look to the maps are all contained in stylesheets. These stylesheets should be included in all UI Maps.

```
...
            <link rel="stylesheet" type="text/css" href="cisDisabled.css"/>
...
```

### Title

Each UI Map should have a <title> tag.

```
...
            <title oraMdLabel="ADDRESS_LBL"></title>
...
```

This will give the UI Map a descriptive title.

- If the UI Map is presented in a "pop-up", the title will be in the window title bar.
- If the UI Map is presented in the page area, the title will be added as a  tag to the UI Map and will appear at the top of the UI Map.
- If the UI Map is presented as a zone map, it will be ignored. The <title> tag should still be included in the HTML as standard.

### Zone Maps

When the map is presented in a zone as part of a portal, the UI Map should have a border so that the information is "contained" within the zone.

```
...
            <body class="oraZoneMap">
...
```

### Page Area Maps vs Pop-Up Maps

The presentation of the UI Maps can vary from design to design. The following standards have been applied to decide when to use a Page Area UI Map and when to use a Pop-Up Map:

- If there are multiple UI Maps in the sequence, always use the Page Area.
- If the UI Map has many input fields, always use a Page Area.
- If the UI Map is a "confirmation" type dialog or only has one or two input fields, use a Pop-Up.

> **Note**
>
> The difference between "just a few input fields" and "many input fields" can be discretionary. The final decision should rest with the dialog designer.


Copyright © 2026, Oracle and/or its affiliates.

### Error Messages

Input maps have a ability to present error messages to the User.

```
...
<table width="100%" cellpadding="12">
     <tr class="oraErrorText">
         <td><a href="" onclick="oraShowErrorAlert(); return false;">
        <span class="oraErrorText" oraErrorVar="ERRMSG-TEXT"></span></a></td>
     </tr>
</table>
...
```

This HTML structure provides the provides the necessary elements and functions to display errors to the User. It should be directly after the <body> tag. When there is no error, nothing will be visible on the UI Map. It will be made visible if an error occurs and the UI Map is re-presented to the User. Clicking on the link (when visible) will result in a pop-up alert appearing with the long error message text.

### Standard Layout and Styles

The information is presented on the UI Map by using a <table> to organize the information in rows and columns.

```
...
<table cellspacing="4" width="100%">
<colgroup>
<col class="oraLabel oraTableLabel" />
             <col class="oraNormal oraTableData" />
             </colgroup>
...
```

The <colgroup> and <col> tags allow for the application of classes to the columns (the label is in the first column and the data is in the second column.). Using these tags mean that the class attribute (to apply styles) does not need to be defined on every <td>.

### Grids (Tables of Data)

A UI Map could contain information that is best presented as a grid. These are referred to as "Embedded Tables". The embedded table can be used to display information or input information.

### Example Embedded Table HTML

The embedded table will be included within a row (<tr>) of the basic layout:

```
...
<tr>
    <td colspan="2" class="oraEmbeddedTable">
        <table oraList="contactInformation" cellspacing="2">
            <thead >
                <tr>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/type></span>
                    </th>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/number"></span>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    </th>
                </tr>
            </thead >
            <tbody>
                <tr>
                    <td class="oraNormalAlt oraDisplayCell">
                        <span oraField="type"></span>
                    </td>
                    <td class="oraNormal oraDisplayCell">
                        <span oraField="number"></span>
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
...
<xml>
    <root>
        <address> 123 Main St</address>
        <city>Alameda</city>
        <state>CA</state>
        <zip>94770</zip>
        <contactInformation>
            <type>Home Phone</type>
            <number>510-555-2287</number>
        </contactInformation>
        <contactInformation>
            <type>Cell Phone</type>
            <number>510-555-4285</number>
        </contactInformation>
    </root>
</xml>
```

**Embedding the Table**

The embedded table is included within the overall table structure. The colspan attribute ensures that the embedded table can span the standard two columns of the overall layout table.

```
...
<tr>
    <td colspan="2" class="oraEmbeddedTable">
    ...
    ...
    ...
    </td>
</tr>
...
```

**Embedded Table Structure**


Copyright © 2026, Oracle and/or its affiliates.

The embedded table is very similar to the basic layout table.

```
...
<table oraList="contactInformation" cellspacing="2">
<thead>
          ...
          ...
</thead>
<tbody>
          ...
          ...
</tbody>
</table>
...
```

- The <table> tag has a slightly smaller cellspacing and it defines the "list" element contained in the XML that will be used to provide the data.
- The <thead> element is used to give the embedded table headings for the columns.
- The <tbody> element is the element that will be repeated for each referenced "list" element in the XML. In the previous example, there are two "contactInformation" list elements, so the displayed embedded table will have two rows.

**Column Headings**

Embedded tables should have headings for the displayed columns. The <thead> tag defines these.

```
...
<thead>
    <tr>
        <th class="oraGridColumnHeader" nowrap="nowrap">
            <span oraLabel="contactInformation/type></span>
        </th>
        <th class="oraGridColumnHeader" nowrap="nowrap">
            <span oraLabel="contactInformation/number"></span>
        </th>
    </tr>
</thead>
...
```

The "nowrap" attribute prevent the column heading from taking multiples lines. If multiples lines are required, the "nowrap" may be removed.

**Input Fields**

Embedded tables may be used for input as well as display only. The framework provides a convenient control to assist in the creation of editable embedded tables.

```
...
<tr>
    <td colspan="2" class="oraEmbeddedTable">
        <table oraList="contactInformation" cellspacing="2">
            <thead >
                <tr>
                    <th class="oraGridColumnHeaderButton"></th>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <th class="oraGridColumnHeaderButton"></th>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/type></span>
                    </th>
                    <th class="oraGridColumnHeader" nowrap="nowrap">
                        <span oraLabel="contactInformation/number"></span>
                    </th>
                </tr>
            </thead >
            <tbody>
                <tr>
                    <td oraType="addGridRow"></td>
                    <td oraType="deleteGridRow"></td>
                    <td>
                        <input type="text" oraField="type"/>
                    </td>
                    <td>
                        <input type="text" oraField="number"/>
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
...
```

There are two new columns added to the input embedded table.

- oraType="addGridRow" will add a "+" button to the row. This will allow the User to add an additional row to the existing grid.
- oraType="deleteGridRow" will add a "-" button to the row. This will allow the User to delete an existing row from the grid.

> **Note**
>
> The <thead> tag also requires these two new columns to be added.

These controls are, as standard, placed at the beginning of the row in the order shown. Either of the controls may be omitted if required (if, for example, Users are not permitted to delete information).

The presence of either of these controls will activate the "empty list" process. This means that if the XML has no data for the "list" specified, the input grid will display with an empty row ready for the input of new information.

### Action Buttons

#### Example Action Button HTML

Action buttons are used to perform some specified function from the UI Map. The actions are as varied as the information being displayed/updated. Below are two common examples:

- Save. Normally used on an Input UI Map to allow a User to save any changes they have made.


Copyright © 2026, Oracle and/or its affiliates.

- Cancel. Normally used on an Input UI Map to allow a User to cancel changes in progress.

```
...
<tr>
  <td colspan="2" class=" oraEmbeddedTable">
    <table cellspacing="2">
      <tr>
        <td>
          <input class="oraButton" oraMdLabel="C1_SAVE_LBL" type="button"
              onClick="oraSubmitMap('OK');"/>
        </td>
        <td>
          <input class="oraButton" oraMdLabel="CANCEL_LBL" type="button"
                          onClick="oraSubmitMap('CANCEL',false);"/>
        </td>
      </tr>
    </table>
  </td>
</tr>
...
```

### Button Standards

The following points highlight some standards related to buttons.

- Buttons are included as an embedded table.
- Buttons should be grouped together. They should not be placed in different areas of the UI Map.
- The location of the buttons depends mainly on the type of UI Map.
  - Display Only UI Maps should have a Record Actions section in the upper right section of the UI map.
  - Input UI Maps should have the buttons at the foot of the UI Map (after all input fields).

#### Available Styles

Styles are all contained in the referenced CSS stylesheets. They are applied by the HTML "class" attribute. The actual style settings used are not documented here as they may be adjusted. This section only specifies when a particular style should be used.

> **Note**
>
> The "class" attribute may reference more than one style (class="oraLabel oraSectionEnd")


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
|---|---|---|
| oraButton | Applied to <input> elements where the type is button. | ...<br><input class="**oraButton**"<br>oraMdLabel="CANCEL_LBL"<br>type="button"<br>onClick="oraSubmitMap('CANCEL',false);"/><br>... |
| oraDisplayCell | Applied to the <td> tag of an embedded table. It defines how the table cell looks (not the data contained inside the cell). | ...<br><td class="**oraDisplayCell**"><br><br></td><br>... |
| oraEmbeddedTable | Applied to the <td> tag that will contain the embedded table. | ...<br><tr><br><td colspan="2" class=" oraEmbeddedTable"><br><table cellspacing="2"><br>...<br>...<br>...<br></table><br></td><br></tr><br>... |

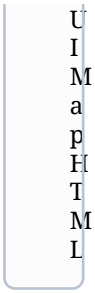

Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraError` | This style is applied to elements that are identified as "error elements". Refer to Display Errors for more information. | |

**Note**

This style is not normally applied directly in the


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
|---|---|---|
| | |   |
| oraErrorText | This style is applied to the elements concerned with error messages. | ```<br>...<br><table width="100%" cellpadding="12"><br>  <tr class="oraErrorText"><br>  <td><a href=""<br>  onclick="oraShowErrorAlert();<br>  return false;"><br>  <span class="oraErrorText"<br>  oraErrorVar="ERRMSG-TEXT"></span><br>  </a><br>  </td><br>  </tr><br></table><br>...<br>``` |
| oraGridColumnHeader | This style is applied to the <td> tags that define column headers within embedded table. | ```<br>...<br><thead ><br>  <tr><br>  <th class="oraGridColumnHeaderButton"></th><br>  <th class="oraGridColumnHeaderButton"></th><br>  <th class="**oraGridColumnHeader**"<br>  nowrap="nowrap"><br>  <span oraLabel="contactInformation/type><br>  </span></th><br>  <th class="**oraGridColumnHeader**"<br>  nowrap="nowrap"><br>  <span oraLabel="contactInformation/number"><br>  </span></th><br>  </tr><br></thead ><br>...<br>``` |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraGridColumnHeaderButton` | This style is applied to the <td> tags that define the column headers for the "+" and "-" buttons used on editable embedded tables. | ... <thead > <tr> <th class="**oraGridColumnHeaderButton**"> </th> <th class="**oraGridColumnHeaderButton**"> </th> <th class="oraGridColumnHeader" nowrap="nowrap"> <span oraLabel="contactInformation/type> </th> <th class="oraGridColumnHeader" nowrap="nowrap">  </th> </tr> </thead > ... |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraInput` | This style is applied to input fields:<br>• <input type="text"><br>• <input type="checkbox"><br>• <select><br>• <textarea> | ```<br>...<br><input type="text" class="oraInput"<br>  oraField="address"/><br>...<br>``` |

> **Note**
>
> This can normally be omitted as input styles are

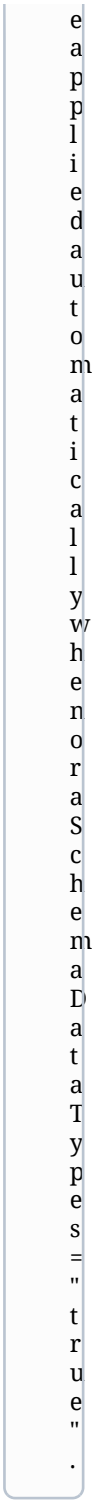
| Style | Comments | Example |
| --- | --- | --- |
| | | appliedautomaticallywhenoraSchemaDataTypes="true". |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraInputMoney` | This style is applied to input fields:<br>• <input type="text"><br>• <select> (rare)<br>• <textarea> (not recommended) | `...`<br>`<input type="text" class="oraInputMoney"`<br>`oraField="amount"/>`<br>`...` |

> **Note**
>
> This cannormallybeomittedasinputstylesar


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| | |  |




Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraInputNumber` | This style is applied to input fields:<br>• <input type="text"><br>• <select> (rare)<br>• <textarea> (not recommended) | `...`<br>`<input type="text" class="oraInputNumber"`<br>`  oraField="count"/>`<br>`...` |

> **Note**
>
> This can normally be omitted as input styles ar


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |

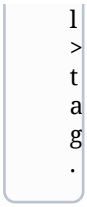
```
appliedautomaticallywhenoraSchemaDataTypes="true".
```


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraLabel` | This style is applied to standard label fields that are right aligned. | ```<br>...<br><td class="oraLabel"<br>  oraLabel="address"></td><br>...<br>``` |

**Note**

This can normally be omitted as it is applied by the <co

| Style | Comments | Example |
| --- | --- | --- |
| | |  |
| `oraLabelAlt` | This style is applied to standard label fields only if it is desired to have the label aligned to the left. | `...`<br>`<td class="`**`oraLabelAlt`**`"`<br>`  oraLabel="address"></td>`<br>`...` |
| `oraLabelCenter` | This style is applied to standard label fields only if it is desired to have the label aligned in the center of the cell. | `...`<br>`<td class="`**`oraLabelCenter`**`"`<br>`  oraLabel="address"></td>`<br>`...` |
| `oraLink` | This style is applied to foreign key references (links). This is automatically added by the UI Map framework but can also be used manually if desired. | `...`<br>`<td class="oraLabel">`<br>`  <a href="www.google.com"`<br>`  class="oraLink">Google</a></td>`<br>`...` |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| **oraNormal** | This style is applied to standard data fields (display only) that are left aligned. | ```<br>...<br><td class="oraNormal"<br>  oraField="address"></td><br>...<br>``` |

**Note**

This can normally be omitted as it is applied by the <c

| Style | Comments | Example |
|---|---|---|
| | | ol>tag. |
| oraNormalAlt | This style is applied to standard data fields (display only) only if it is desired to have the data aligned to the right. | ...<br><td class=" **oraNormalAlt**"<br>oraField="address"></td><br>... |
| oraNormalCenter | This style is applied to standard data fields (display only) only if it is desired to have the data aligned in the center of the cell. | ...<br><td class=" **oraNormalCenter**"<br>oraField="address"></td><br>... |

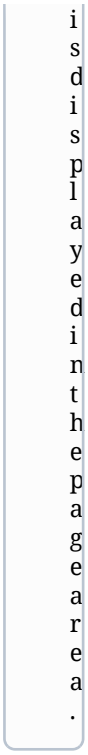

Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraPageTitle` | This style is applied to the element that contains the page title. | `...`<br>`<span class=" oraPageTitle"`<br>`  oraMdField="PAGE_TITLE_LBL"></span>`<br>`...`<br><br>**Note**<br>This style is not normally applied directly in the UI M |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| | | apHTML.TheiscreatedbytheUIMapframeworkwhenthetheUIMap |


Copyright © 2026, Oracle and/or its affiliates.

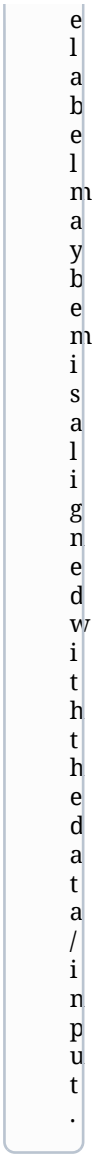
| Style | Comments | Example |
| --- | --- | --- |
| | |  |




Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
|---|---|---|
| `oraSectionEnd` | This style is applied to the <td> tags at the end of a "section" (group of elements). It provides some space to separate the section from the following information. | `...`<br>`<tr>`<br>`  <td class="oraSectionEnd"`<br>`  oraLabel="zip">`<br>`  </td>`<br>`  <td class="oraSectionEnd"`<br>`  oraField="zip">`<br>`  </td>`<br>`</tr>`<br>`...`<br>**Note**<br>The style must be applied to both <td> tags or th |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| | |  |



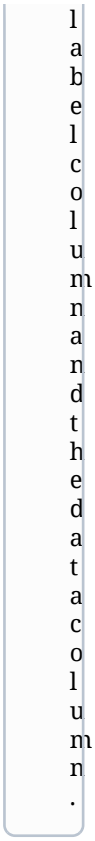

Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraSectionHeader` | This style is applied to the <td> tag used to give a heading for a section within the information being displayed. It does not provide spacing before or after itself. The oraSectionStart and oraSectionEnd classes are used for this. | ...<br><tr><br><td class="**oraSectionHeader**"<br>colspan="2"<br>oraMdField="DATA_SECTION_LBL"></td><br></tr><br>... |

**Note**

The sectionheader should span both the


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| | |  |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `oraSectionStart` | This style is applied to the <td> tags at the start of a "section" (group of elements). It provides some space to separate the section from the previous information (often a section header). | ```<br>...<br><tr><br>  <td class="oraSectionStart"<br>   oraLabel="zip"></td><br>  <td class="oraSectionStart"<br>   oraField="zip"></td><br></tr><br>...<br>``` |

**Note**

The style must be applied to both <td> tags or


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| | | the label may be mis aligned with the data/input. |
| `oraTableData` | This style is applied to the <col> tag for the data column of the main table (second column). It is used to provide a percentage width for the horizontal space to be used for the information. | ```<br>...<br><colgroup><br>  <col class="oraLabel<br>  oraTableLabel"/><br>  <col class="oraNormal<br>  **oraTableData**"/><br></colgroup><br>...<br>``` |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
|---|---|---|
| oraTableLabel | This style is applied to the <col> tag for the label column of the main table (first column). It is used to provide a percentage width for the horizontal space to be used for the labels. | ...<br><colgroup><br><col class="oraLabel **oraTableLabel**" /><br><col class="oraNormal oraTableData"/><br></colgroup><br>... |
| oraTinyText | This style is typically applied directly beneath an <input> tag to provide information or a hint to the user concerning information relevant to the input. For example, name or address format. | ...<br><tr><br><td oraLabel="address"></td><br><td><br><input type="text" oraField="address"/><br> <br></td><br></tr><br>... |
| oraZoneMap | This style is used applied when the UI Map is to be displayed as a zone on a portal. | ...<br><body class="oraZoneMap"><br>... |
| ou-align-items-after-text | Tree nodes may be designed to have their node text directly followed by icons and badges. In some use cases there is a need to align those icons and badges in a column after the text. This is especially true for "narrow" trees where the text tends to wrap. This style aligns tree node icons and badges to the right in a column down the tree. | <ou-tree treename="..." class="ou-align-items-after-text"></ou-tree> |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `ou-clear-hyperlink-color` | This style may be used to override the standard hyperlink color and use the default text color instead. It applies to all the hyperlinks in the defined container. | `<body class="ou-clear-hyperlink-color"> - Applied to all hyperlinks on the entire page`<br>`<div class=ou-clear-hyperlink-color"> - Applied to all hyperlinks in the specified <div>`<br>`<a class="ou-clear-hyperlink-color"> - Applied to a single element` |
| `ou-icon-color-standard`<br>`ou-icon-color-gray`<br>`ou-icon-color-red`<br>`ou-icon-color-green`<br>`ou-icon-color-paleblue`<br>`ou-icon-color-yellow`<br>`ou-icon-color-orange`<br>`ou-icon-color-white`<br>`ou-icon-color-reverse-video` | These styles may be used to change the color of an icon. | `...`<br>`<ou-button type="image" imgref="F1NEWWIN" ...`<br>`class="ou-icon-color-gray" ...>`<br>`</ou-button>`<br>`...` |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
| --- | --- | --- |
| `ou-icon-small`<br>`ou-icon-xsmall` | These styles may be used to change the size of the iconography.<br>u-icon-small makes the icon 16x16<br>ou-icon-xsmall makes the icon 12x12.<br>This style should only be used for decorator icons. Clickable icons should be standard size for accessibility. | `...`<br>`<ou-decorator imgRef="F1ACCT" class="ou-icon-small"></ou-decorator>` |
| `ou-show-as-grid` | This style renders single elements as if they were rows in a grid. | `<div class="ou-show-as-grid">`<br>`<div>`<br>`  <label .../>`<br>`  <span .../>`<br>`</div>`<br>`...`<br>`<div>`<br>`  <label .../>`<br>`  <span .../>`<br>`</div>`<br>`</div>` |
| `ou-show-as-label-above` | This style renders an element in redwood view mode which sets the label iabove the field value. The style is ignored in redwood view mode. | `<div class="ou-show-as-label-above">`<br>`  <label .../>`<br>`  <span .../>`<br>`</div>` |
| `ou-show-as-label-in-front` | This style renders an element in classic view mode which sets the label in front of the field value. The style is ignored in classic view mode. | `<div class="ou-show-as-label-in-front">`<br>`  <label .../>`<br>`  <span .../>`<br>`</div>` |


Copyright © 2026, Oracle and/or its affiliates.

| Style | Comments | Example |
|---|---|---|
| `ou-show-as-standard-lines` | This style may be used to render a simple name/ value two columns table as single elements. It "converts" the table into standard looking line elements. | `<table class="ou-show-as-standard-lines"><tr>`<br>`<td>... name ...</td>`<br>`<td>... value ...</td>`<br>`</tr></table>` |

#### Using OJET

There are some UI maps delivered by the product that use UI widgets provided by Oracle JavaScript Extension Toolkit (OJET). Releases for OJET do not always align with releases of the framework. In addition, there are times when OJET adjusts APIs that the product uses. The framework will attempt to ensure that each release of the product has the latest and greatest version of the OJET libraries. Implementations are discouraged from attempting to use features in OJET that are not used by the product because the product is not necessarily testing those features and is not ensuring that upgrades to the APIs for those features are backward compatible.

Note that the product isolates the references to OJET into a UI map fragment that is included in the maps that use OJET widgets. This is so that changes to future versions of OJET are minimized to a single place. The map is called F1–OJETLIBS. If your implementation wants to use OJET, the recommendation is to use this UI map fragment.

# Process Flows

This section describes concepts and provides guidelines related to the configuration of various type of process flows.

- Understanding Process Flows
- Designing Process Flows
- Defining Process Flow Types

## Understanding Process Flows

A process flow is a user interface guiding a user through a series of actions in order to accomplish a specific task. The task can be as simple as the collection of information in order to update business data or involve more complex logic such as submitting and tracking batch processes, exchanging messages with an external system, etc.

This section describes topics related to designing and working with process flows.

#### A Process Flow Is Made Of Panels

Each process flow consist of a number of sequential "steps" needed to accomplish a certain task. Each step is represented by a stop on a progress bar and an associated panel.

Progress is linear in that each step may only lead to a single next step. A process flow always starts at a single initial step but allows for one or more final steps at which the user may choose to complete the process.


Copyright © 2026, Oracle and/or its affiliates.

At any step the user may take the following actions:

- Enter data or take action as prompted by the step's panel.
- Continue to a next step, if any.
- Navigate back to any step they have previously visited.
- Finish the process flow if the current step is a final step.
- Save off their work on the process flow and either continue their work or navigate away. At a later time the user may resume their work on the process flow from where they have left off.
- Cancel the process flow.

A process flow type defines the entire metadata needed to control the behavior of process flows of a given type. This includes the sequence of steps, the panel and rules associated with each step and more.

### Panel Presentation

The panel presented on each step of the process flow is rendered based on the panel type selected on the step configuration.

For a **Data Area** type of panel, the panel is rendered using UI Hints built into the schema of a specified data area.

For a **UI Map** type of panel, the panel is rendered using a specified UI map fragment.

For a **Panel Set** type of panel, the panel is rendered as a set of tabs, one for each panel that references the current panel as its parent panel. Clicking on each tab renders the UI based on the selected panel's type. The behavior of a panel set is analogous to or can be thought of as a nested process flow within a single panel. A panel set may not include another panel set.

### A Single UI Map

While the user enters data one panel at a time, data is collected and captured at the entire process flow level. All the elements edited and displayed on the various panels of the process flow must be included in the overall process schema and referenced by their appropriate XPath location in that schema. Each type of process flow defines its unique schema on a designated UI map.

The UI map's HTML should simply include the **Process Flow Controller (F1-ProcessController)** UI map fragment and nothing else. The latter is designed to render the user interface for a process flow based on the metadata defined on its process flow type.

### Panel Scripts

A single **Pre-Processing** service script may be associated with a panel for the purpose of preparing the data before the panel is presented to the user. The system calls the script each time the user navigates to this panel on a non-finalized process flow.

A single **Post-Processing** service script may be associated with a panel for the purpose of validating the data entered by the user on that panel. The system calls the script each time the user exits the panel on a non-finalized process flow.

### Process Flow Scripts

A single **Process Start** service script may be associated with a process flow for the purpose of preparing the data before a newly initiated process flow is presented to the user.


Copyright © 2026, Oracle and/or its affiliates.

A single **Process End** service script is executed when the user clicks on the **Finish** or **Cancel Process** buttons to complete the process flow. The entire set of data collected by the process flow UI map schema is provided to this service script for final processing.

> **Note**
>
> The schema of these scripts should be the same as the **Process Flow UI map** schema. This would ensure that business rules have access to the entire data captured by the user.

### Summary Panel

By default, when a user finishes a process flow, as well as when they view an already closed process, the process flow portal displays the last panel the user was working on. For more complex process flows, you may provide a panel that summarizes information from the entire process. When specified on the process flow type, the system displays that summary panel instead a the default view of a closed process. The user may toggle between the summary view and the detailed view at any time.

The summary panel may be implemented as a UI map fragment or a data area like other panels but may not be associated with any panel script.

### Launching A New Process Flow

Each process flow must be initiated and executed by a designated BPA script that has the following simple steps:

- Set BPA area height to zero.
- Populate the temporary variable **$processCode** with the code of the specific process flow type.
- Transfer control to the **Process Flow Navigation (F1-PROCEXEC)** common BPA script.

Launching a process flow of a specific type is assumed to be made from a designated menu option or within a specific context applicable to that type of work. As such, this item needs to be configured specifically for each type of process flow. The navigation option associated with that launching option should reference the BPA script associated with the corresponding type of process flow.

### Saving Off Work

The user may save off their work on a process flow at any time. If enabled on the process flow type, the system would also automatically save the data when the user navigates to another panel or away from the process flow portal.

On either manual or automatic save operations, the **Process Manager** script defined on the process flow type is called to save off the entire process flow data on a designated record in the database.

The base product provides a generic **Process Flow** maintenance object that is designed to support the storage of any type of process flow as well a corresponding process manager script. Should process flows of a specific type be stored elsewhere, then a dedicated process manager script should be implemented and used instead.


Copyright © 2026, Oracle and/or its affiliates.

### Review Process Flow Records

A process flow record is created when a user saved off their work or the process flow type enables automatic saving. When the user finishes a process flow, depending on the process flow type configuration the record may be deleted or retained for audit purposes.

The **Process Flow Query** portal allows a user to resume their work on a in-progress flow and review the information associated with any finalized flows.

Reviewing process flow records of a specific type could also be made from a designated menu option or within a specific context applicable to that type of work. As such this item needs to be configured specifically for each type of process flow. The navigation option associated with that item should reference a query portal designed to query and manage the specific type of process flow records.

## Designing Process Flows

A wide range of process flows may be designed to implement online tasks business users may need to perform. Refer to Understanding Process Flows for more information.

Below is a high-level summary of the steps required to design and configure various types of process flows.

### Analyze The Task

The following provides a the high-level guidance in analyzing the necessary data and rules needed to support an implementation of a new type of process flow:

- Identify the data and business rules that are needed to complete the entire task.
  - When possible, organize data into logical groups of details defined as data areas.
  - Design logic for the final processing script that accepts the entire data entered along with the user's action to either complete or cancel the task.
  - If data needs to be initially prepared for a newly initiated process flow, design logic for a corresponding initial processing script.
  - Determine whether a summary panel may be useful for the user to review the outcome of a completed process.
- Design the sequence of panels the user should follow to complete the task. Identify the initial and final panels.
- Review each panel and determine the following:
  - Is it a single panel or a nested process flow (i.e. a panel set)?
  - What is the title of the panel and its label on the progress bar?
  - How should the layout be rendered? Use a data area that leverages UI Hints to describe the panel layout when possible. For more complex HTML requirements you may use a UI map fragment.
  - Should details on this panel be defaulted with some initial values when the user enters this panel? If so, design logic for a corresponding pre-processing script.
  - If data entered on this panel should be validated before the user exits the panel, design logic for a corresponding post-processing script.
- Design the summary panel if needed.


Copyright © 2026, Oracle and/or its affiliates.

- Determine whether this type of process flow should be automatically saved by the system or solely rely on the user to control when data is saved.
- Determine whether a record of a completed process flow should be retained for audit purposes or deleted upon completion.
- Determine whether a user may delete in progress records of such process flow.
- Determine whether process flows of this type are saved and stored in a dedicated table in the database or are saved in the generic table provided by the base product. The assumption is that introducing new designated storage maintenance objects is not common. Refer to your edge product documentation for additional maintenance objects that may have been provided to support specific types of process flows shipped with the product.
- Determine the context in which a user can initiate, resume work on, and review process flows of that type.
  - The process flow may be launched from a menu or a favorite script etc.
  - Design the query portal a user would use to find and resume a saved off process flow of this type as well as review completed process flow records. This step is only needed if the base product query portal is not suitable for this type of tasks. Depending on the context of this task, the query may be associated with the same menu line that initiates the process flow.
  - Determine if specific characteristics may need to be stored along with the process flow record for the purpose of identifying in-progress flows by context. Design your process flow service scripts to populate these characteristic values in the control data area available to these scripts.

**Using UI Map Panels**

Typically a map fragment is rendered as part of a complete UI map but when used as a process flow panel the system renders the panel assuming the fragment contains a complete map.

Therefore, such fragment should includes the following items:

- The standard error message section used for reporting errors.
- This line <xml style="display:none"></xml> at the end of the HTML content for proper binding of data to their HTML elements.

A UI map based panel allows for more control over some of the buttons available to the user. For example, the panel may prevent the user from proceeding to the next panel unless required information is entered. Refer to the explicit APIs listed in the process controller UI Map (F1-ProcessController) for more information.

**Configure The Process Flow Type**

The following describes the high-level steps needed to configure a new type of process flow:

- Create the UI map for the process flow. Its schema should contain all the details needed to complete the task. Its HTML should only include the process controller UI map fragment.
- Create a start processing service script if needed.
- Create a final processing service script such that it applies all the rules needed to complete or cancel the task.
- For each panel perform the following:


Copyright © 2026, Oracle and/or its affiliates.

  - Create a UI map fragment for it unless it corresponds to a data area already included in the process flow schema. Make sure the XPath of each element references in the data area or UI map is the same XPath of that element in the process flow's UI map schema.
  - Create a pre-processing service script if needed. Not applicable to the summary panel.
  - Create a post-processing service script if needed. Not applicable to the summary panel.
  - Set up a label and title MD fields if existing ones cannot be used.
- Create a BPA script for the process flow.
- Configure the process flow type record.
- Set up a navigation option and reference the BPA script on it. Associate the navigation option with a menu or provide other means for the user to initiate the process flow.
- Create a query portal for the specific type of process flow records. Add the new query portal to a menu or provide other means for the user to resume work on a process flow they have saved off or review completed records.

**Control Visibility of Standard Buttons**

You may leverage the following javaScript functions to control the visibility of standard process flow buttons:

- Next: `enableNext()`, `disableNext()`, `hideNext()` , `showNext()`.
- Finish: `enableFinish()`, `disableFinish()`, `hideFinish()` , `showFinish()`.
- Save: `enableSave()`, `disableSave()`, `hideSave()` , `showSave()`.

## Defining Process Flow Types

This portal is used to maintain process flow types.

Refer to Understanding Process Flows for more information.

You can access the portal from the **Admin Menu**, and then **System**, and then **Process Flow Type.** You are brought to a query portal with options for searching for a specific process flow type. Once a process flow type has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Process Flow Type.** This zone provides information about the selected process flow type.

# Action Method

Action method is an object that may abe used to define configuration based on some criteria. For example, when initiating a Start Service scenario, the service request type may differ based on the CIS division and the customer class. When a user initiates a Start Service process, the system can prompt the user for information about the customer that can determine the CIS division and the customer class. The system can then call an algorithm passing this information to retrieve the appropriate service request type to use as defined in the action method. Throughout this section, the term 'criteria' is used to identify the information used to determine the desired configuration and the term 'details' or 'results' is used to identify the configuration information being determined.

If you were to design a use case to use this functionality consider the following:


Copyright © 2026, Oracle and/or its affiliates.

- What information needs to be identified (the 'details'/ 'results')? The example above uses a configuration object as its example (service request type). However, the information you are determining could be anything that can differ by some combination of attributes. You could use this to determine which collection algorithm to execute based on criteria. Or the number of days to wait to continue with some process for some use case.
- When and where do I need to determine this information and what do I do with it once I receive it?
  - Is this needed as part of a user interaction?
  - Or is this needed for internal processing?
- What is the criteria needed to determine the desired information? Do I have this information at the time that I need it? Or do I need to prompt a user for the information?

The framework product supplies the generic objects needed to support this type of functionality. The individual edge products or implementations can build their use cases with the tools provided. The topics in this section further describes the provided features.

### Action Method Role

The action method functionality is driven by the Action Method Role, which represents a given use case. Some examples of action method roles that may be relevant in an enterprise product are 'Start Service', 'Stop Service' and 'Transfer Service'. The valid action method roles are defined using an extendable lookup and referenced on the Action Method. One and only one action method may be defined for one action method role.

The action method role record may reference an action method role BPA script, if the use case for this Action Method Role involves user interaction. The responsibility of this script, if needed is to take the action method role value, determine the action method that references the role and identify the more specific Action Processing BPA script (defined on the business object) and transferring control to that script. This reference may be used for audit or information purposes, depending on how you have implemented the user experience. The action method role BPA is ideally configured on an appropriate menu entry or an action button on a specific user interface related to this use case.

If your business use case does not require user interaction, then an action method role BPA script is not necessary.

### Action Method Business Object

The action method's business object is used to define the information that needs to be configured for both the criteria and the results. Continuing to use the above example, the business object for a Start Service use case defines a list that captures the CIS division and customer class along with the service request type to use. The information is configured in the XML data area for the action method and the schema design is dictated by the business requirements. For example, it may be that your use case dictates a default value for the results (in our case, the service request type) plus override values based on a division and customer class combination.

### Get Action Method Details Plug-in

Besides defining the schema information for capturing the criteria and results, the business object defines the algorithm that should be invoked to retrieve the results based on the criteria.

The plug-in spot is **Get Action Method Details**. Its API is flexible in the information it receives and returns so that each action method use case can design the algorithm based on its business requirements.


Copyright © 2026, Oracle and/or its affiliates.

The algorithm receives a list of 'Criteria Data' using a 'name' and up to 5 values (to handle multi-part primary keys). Typically only 'value 1' is needed for each entry. The algorithm type designed for a given action method business object can determine what information is receives. It could be that it expects the criteria elements defined on the action method. In our example, the algorithm type could expect to receive the CIS division and customer class. Alternatively, the algorithm type could be designed to receive a value that can be used to determine the criteria needed. For example, it could receive the account ID and use that information to determine the CIS division and customer class.

The algorithm returns one or more results as 'Details'. The information returned is based on the use case. In our example, the service request type is returned. Your business requirements may use the action method role to return multiple results, for example a service request type and a process flow type to invoke to capture all the information needed to start service.

The product provides a business service `F1-RetrieveActionMethodDetails` to call the algorithm for a given action method value.

#### Action Processing BPA

If your business requirement requires user interaction either to determine the criteria to pass to the Get Action Method Details algorithm or to use the resulting information to continue with an online action, an Action Processing BPA is needed. This BPA should be designed and linked to your action method business object as an option using the **Action Processing BPA Script** option type.

The design of this BPA depends on your business requirements. But at a high level, it may need to prompt the user for information needed to determine the criteria associated with the action method details, then call the `F1-RetrieveActionMethodDetails` business service to execute the algorithm for your action method. Once the results are returned, the script would proceed with the next step in the business use case.

> **Note**
>
> If any additional information needs to be gathered by the server before or after the call to the business service that executes the algorithm, then the recommendation is that the BPA should call a service script that does all the server level logic at one time.

#### Action Method Role vs. Action Method Business Object

The action method role defines the use case. The business object is used to design the schema for the criteria and the results and identifies the action processing BPA and the Get Action Method Details plug-in. It may warrant a unique business object for each action method role. However, there may be cases where more than one action method role and its action method can reuse the same business object. For example, imagine there are different service request types for Start Service, Stop Service and Transfer Service, however in all cases the valid value is driven by the CIS division and customer class. These three separate action method roles and action methods can reuse a common business object defining the criteria of CIS division and customer class determining the appropriate service request type.

- Defining Action Methods

# Defining Action Methods

Refer to Action Method for an overview of action method functionality.


Copyright © 2026, Oracle and/or its affiliates.

To maintain the action methods applicable to your implementation, open **Admin**, and then **System**, and then **Action Method.**

This is a standard All-in-One portal and includes the standard List and display zones for an action method.

The information captured on the action method depends on the business objects supported by your product or implementation. Refer to the embedded help text for more information.

# Algorithms

In this section, we describe how to set up the user-defined algorithms that perform many important functions including:

- Validating the format of a phone number entered by a user.
- Validating the format of a latitude/longitude geographic code entered by a user.
- In products that support payment and billing:
  - Calculating late payment charges.
  - Calculating the recommended deposit amount.
  - Constructing your GL account during the interface of financial transactions to your GL
- And many other functions...
- The Big Picture Of Algorithms
- Algorithm Entity
- Defining Algorithm Types
- Defining Algorithms
- Advanced Algorithm Tips

## The Big Picture Of Algorithms

Many functions in the system are performed using an algorithm. Algorithms are called at strategic points (often referred to as a "plug-in spot") within system processing to perform specific code. The plug-in spots are preconfigured with the product and vary from object to object. Algorithms are used by the base product to provide out of the box functionality. The algorithm framework provides a means for implementations to insert their own custom code, which can be instead of the base delivered code or in addition to it.

> **Note**
>
> **Algorithm = Plug-in.** We use the terms plug-in and algorithm interchangeably throughout this documentation.

### Algorithm Type Versus Algorithm

Often the documentation will just use the term 'algorithm', but in actuality there are two important objects that provide this plug-in functionality.

- An **Algorithm Type** defines program to execute and identifies the plug-in spot. The program may be defined using Java or using a plug-in script. The algorithm type may also define parameters that must be supplied to algorithms of this type.


Copyright © 2026, Oracle and/or its affiliates.

- An **Algorithm** is an instance of an **Algorithm Type**. If the algorithm type does not define any parameters, then there will be only one algorithm for the algorithm type. If the algorithm defines parameters, then there could be several algorithms for a given algorithm type, one for each unique set of parameter values for a given use case. The algorithm is the object that gets referenced on a given control table when configuring your system.

### Algorithm Plug-in Spot

The plug-in spot, also called Algorithm Entity and System Event, defines several aspects of the functionality for algorithms that are executed.

- When in system processing are algorithms of this plug-in spot executed? For example, algorithms for the Installation Option - To Do Pre-creation are called when a To Do is getting added, before any insert to the database is performed.
- The API for the plug-in spot. This is often referred to as the "hard parameters" for the plug-in spot. What information is every algorithm program provided when called? Additionally, if the algorithm is expected to return information to the caller, this is also part of the hard parameters. For example, the Installation Option - To Do Pre-creation plug-in spot receives all the information about the To Do that is about to be added. It may add or update values. It may also return an indication to not create the To Do, if desired.

  > **Note**
  >
  > The hard parameters are different from the parameters defined on the algorithm type. Those are referred to as the "soft parameters". Those parameters are used to configure settings that the algorithm type code uses per its specific functionality.

- What is the responsibility of algorithms for the plug-in spot? When defining a plug-in spot, the product defines the responsibility of the code for the algorithm types created for that spot. For example, is the algorithm determining information and returning it? Or is it supposed to perform an update? Algorithms for the Installation Option - To Do Pre-creation plug-in spot should not do any updates. Rather they should adjust the To Do information (if desired) in the hard parameters and return them. The calling logic will use that adjusted information when inserting the To Do record.
- Where do you configure the algorithm? We have been using the Installation Option - To Do Pre-creation plug-in spot as an example. As the description suggests, these algorithms are configured on the Installation Option - Algorithm collection. Each plug-in spot defines where (on which control table) the algorithms are plugged in. The system standard for the description of the Algorithm Entity flag value that defines the available plug-in spots is "Control Table Name - Event". For example the To Do Type - Calculate Priority algorithm plug-in spot means that algorithms are configured on the To Do Type. This information also indicates the granularity of the algorithm. Installation Option algorithms are system wide. But an algorithm plugged in on To Do Type, for example, indicates that only To Do Entries for that To Do Type will have that algorithm executed.
- Are multiple algorithms allowed? There are some plug-in spots where it doesn't make sense to support multiple algorithms. An example of this is the Batch Control - Select Records plug-in. This plug-in is used by the plug-in driven batch process and is responsible for selecting the records to process. Only one algorithm of this type is allowed on a given Batch Control. An example of a plug-in spot that would support multiple algorithms is a validation plug-in spot. Note that when a plug-in spot supports multiple algorithms, there can be yet additional aspects of the plug-in spot design.
  - In some cases, this type of plug-in spot simply executes each algorithm. A validation plug-in spot is an example of this.


Copyright © 2026, Oracle and/or its affiliates.

- In some cases, the algorithm could return an indication to the plug-in spot 'driver' to not execute any subsequent algorithms. Business Object Status - Enter is an example of a plug-in spot that supports this capability.
- In some cases, the plug-in spot may support sharing information from one algorithm to the next. Migration Plan - Pre-Compare is an example of a plug-in spot that supports this capability.

Refer to Algorithm Entity for information about a portal provided to display information about a plug-in spot.

**Base Product Algorithms**

The base product provides many, many algorithm types and algorithms out of the box. Sometimes algorithms are delivered to provide support for common use cases that we know our clients will need. Many algorithms, however, are provided because we are implementing system functionality using the same infrastructure and capabilities that our implementation use for their own custom code.

When an algorithm type is delivered, the decision as to whether an algorithm is also provided depends on some factors related to whether soft parameters are defined for the algorithm type.

- If no parameters are defined for the algorithm type, a corresponding algorithm is typically also provided.
- If parameters are defined and the values are on configuration data that is normally owned by the implementation, a base algorithm will probably not be defined.
- If parameters are defined and the values are not dependent on configuration data, it's possible that one or more algorithms are provided.

To review the algorithm types that are delivered with the product, go to the Algorithm Type query. The query provides several search options, including searching by the algorithm entity lookup value.

- How To Add A New Algorithm
- Minimizing The Impact Of Future Upgrades

## How To Add A New Algorithm

Before you can add a new algorithm, you must determine if you can use one of the algorithm types supplied with the system by using the algorithm type query.

If you can use one of the algorithm types, simply add the algorithm and then reference it on the respective control table. Refer to Defining Algorithms for how to do this.

If you have to add a new algorithm type, you may have to involve a programmer. Let's use an example to help clarify what you can do versus what a programmer must do. Assume that you require an additional geographic type validation algorithm. To create your own algorithm type you must:

- Write a new program to validate geographic type in the appropriate manner. Alternatively, you may configure a plug-in script to implement the validation rules. The advantage of the latter is that it does not require programming. Refer to plug-in script for more information.
- Create an Algorithm Type called **Our Geographic Format** (or something appropriate). On this algorithm type, you'd define the name of the program (or the plug-in script) that performs the function. You'd also define the various parameters required by this type of algorithm.
- After creating the new Algorithm Type, you can reference it on an Algorithm.


Copyright © 2026, Oracle and/or its affiliates.

- And finally, you'd reference the new Algorithm on the Geographic Type that requires this validation.

## Minimizing The Impact Of Future Upgrades

The system has been designed to use algorithms so an implementation can introduce their own logic in a way that's 100% upgradeable (without the need to retrofit logic). The following points describe strong recommendations about how to construct new algorithm type programs so that you won't have to make program changes during future upgrades:

- Do not alter an algorithm type's hard parameters. For example, you might be tempted to redefine or initialize parameters defined in an algorithm type's linkage section. Do not do this.
- Follow the naming conventions for the new algorithm type code and your source code, i.e., both the source code and the algorithm type should be prefixed with "CM". The reason for this naming convention is to make it impossible for a new, base-package algorithm type from overwriting your source code or algorithm type meta-data (we will never develop a program or introduce meta-data beginning with CM).
- Avoid using embedded SQL to perform insert/update/delete. Rather, invoke the base-package's object routines or common routines.
- Avoid using base messages (outside of common messages, i.e., those with a message number < 1000) as we may deprecate or change these messages in future releases. The most common problem is caused when an implementation clones a base package algorithm type program because they need to change a few lines of logic. Technically, to be 100% upgradeable, you should add new messages in the "90000" or greater category (i.e., the category reserved for implementation-specific messages) for every message in your new program even though these messages may be duplicates of those in the base package.

# Algorithm Entity

This portal is used to view information about a plug-in spot (algorithm entity). Refer to Algorithm Plug-in Spot for more information.

Navigate using **Admin**, and then **System**, and then **Algorithm Entity**.

Use the **Algorithm Entity Query** portal to search for an existing plug-in spot. Each plug-in spot is defined in the Algorithm Entity lookup (`ALG_ENTITY_FLG`). Once an algorithm entity is selected, you are brought to the viewing portal to view details about the algorithm entity.

The following zones appear on the Main page.

- **Algorithm Entity**. This zone provides information about the plug-in spot, including a link to view the Plug-in Spot API, also called the hard parameters for the plug-in spot. In addition it displays the detailed description, which should provide information about the algorithm's responsibility and when it is called.

  > **Note**
  >
  > This information is also available via a help icon on the algorithm type page and when defining a plug-in script, once you select the algorithm entity value.

- **Algorithm Types**. All existing algorithm types for the plug-in spot are listed. If there are algorithms, the count of algorithms are displayed. Clicking the count opens another zone.


Copyright © 2026, Oracle and/or its affiliates.

- **Algorithms**. If you clicked the algorithm count for an algorithm type, another zone appears with information about the algorithms for that type.

Refer to the embedded help in the portal for more information about each zone.

# Defining Algorithm Types

The system provides many algorithm types to support base product functionality.

You can access the portal from the **Admin**, and then **System**, and then **Algorithm Type**. You are brought to a query portal with options for searching for a specific algorithm type. Once an algorithm type has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to The Big Picture Of Algorithms for more information.

> ⚠ **Caution**
>
> Important! If you introduce a new algorithm type, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Algorithm Type.** This zone provides information about the selected algorithm type.
  - Algorithm Entity defines where algorithms of this type can be "plugged in". The values for this field are customizable using the lookup table. This field name is `ALG_ENTITY_FLG`.
  - > ⓘ **Note**
    >
    > When adding a new algorithm type that is for a Java program, the parameters are automatically generated based on the Java code. Once an algorithm type exists, any additional parameters defined in the Java code should be manually added to the algorithm type. For other program types, algorithm type parameters must be manually defined.
  - > ⓘ **Note**
    >
    > When a new algorithm type parameter is added for any program type, existing algorithms for the algorithm type do not automatically get updated with the new parameter. The algorithms must be manually updated.
- **Algorithms.** This zone lists available algorithms for the selected algorithm type .

### Actions

The page supports the following special actions:

- **View the source.** If the program is shipped with the base package, you can use the adjacent button to display the source code of this program in the Javadocs viewer. For plug-in scripts, drill into the plug-in script to view the details.


Copyright © 2026, Oracle and/or its affiliates.

# Defining Algorithms

This portal is used to view and maintain information about an algorithm.

You can access the portal from the **Admin Menu**, and then **System**, and then **Algorithm**. You are brought to a query portal with options for searching for a specific algorithm. Once an algorithm has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to The Big Picture Of Algorithms for more information.

> ⚠ **Caution**
>
> Important! If you introduce a new algorithm, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Algorithm.** This zone provides information about the selected algorithm.
- **Other Algorithms.** This zone lists other algorithms of the same algorithm type. The zone is only visible when such algorithms exist.

The following zones may appear as part of the portal's **References** tab page

- **Algorithm References.** This zone list various configuration entities referencing the current algorithm.

> ⓘ **Note**
>
> If the product delivers an algorithm with parameter values defined, an implementation may override the base provided parameter values by adding an additional effective dated collection of parameters.

> ⓘ **Note**
>
> If an algorithm is defined and subsequently a new parameter is added to the algorithm type, existing algorithms for the algorithm type should be updated to add a new effective dated entry to the parameter collection. At this point the latest list of parameters for the algorithm type are visible. Configure the parameters accordingly.

# Advanced Algorithm Tips

### Initiating Actions that Cannot be Rolled Back

If you have an algorithm that creates a sync request or performs some other logic that could cause a message to be sent and cannot be rolled back, you should be sure to cater for warnings in the design.

- The algorithm should be performed after all validations that could accumulate warnings.


Copyright © 2026, Oracle and/or its affiliates.

- The algorithm should include a step to call the Business Service `F1-CheckWarning` and see if any warnings have been accumulated. If so, don't perform the action. It means that the user will get a warning and if they click "OK", all the processing will happen again, but this time without accumulating any warnings. Refer to the plug-in script `F1-GnChgDCap` for an example.

# Script Options

We use the term "script" to define processing rules that your implementation sets up to control both front-end and back-end processing:

- Rules that control front-end processing are called Business Process Assistant (BPA) scripts. For example, your implementation could set up a BPA script to assist a user through your organization's payment cancellation process.
- Rules that control back-end processing are called Server-based scripts. For example, your implementation could set up a server-based script to control the processing that executes whenever a given type of adjustment is canceled.

The topics in this section describe how to configure your scripts.

- The Big Picture Of Scripts
- The Big Picture Of BPA Scripts
- The Big Picture Of Server-Based Scripts
- Defining Scripts
- Merging Scripts
- Maintaining Functions

## The Big Picture Of Scripts

This section describes features and functions that are shared by both BPA scripts and server-based scripts.

- Scripts Are Business Process-Oriented
- A Script Is Composed Of Steps
- A Script May Declare Data Areas
- Securing Script Execution
- Script Engine Version

### Scripts Are Business Process-Oriented

To create a script, you must analyze the steps used to implement a given business process. For example, you could create a "stop auto pay" BPA script that:

- Asks the user to search and select the customer / taxpayer
- Asks the user to define the date on which the person would like to stop making automatic payments
- Invokes a server-based script that populates the end-date on the account's latest automatic payment instructions.

After you understand the business process, you can set up a script to mimic these steps. If the business process is straightforward (e.g., users always perform the same steps), the script


Copyright © 2026, Oracle and/or its affiliates.

configuration will be straightforward. If the business process has several logic branches, the composition of the script may be more challenging.

## A Script Is Composed Of Steps

A script contains one or more steps. For example, a "stop auto pay" BPA script might have three steps:

- Ask the user to search and select the customer / taxpayer
- Ask the customer the date on which they'd like to stop making automatic payments (and default the current date)
- Invoke a server-based script that, in turn, updates the account's auto pay options.

Each step references a step type. The step type controls what happens when a step executes. It might be helpful to think of a script as a macro and each step as a "line of code" in the macro. Each step's step type controls the function that is executed when the step is performed.

> **NOT_SUPPORTED**
>
> Refer to How To Set Up Each Step Type for a detailed description of all available step types and how to set them up.

## A Script May Declare Data Areas

Both BPA and server-based scripts may have one or more data areas:

- If the script contains steps that exchange XML documents, you must declare a data area for each type of XML document. For example, if a BPA script has a step that invokes a service script, the BPA script must declare a data area that holds the XML document that is used to pass information to and from the service script.
- You can use a data area as a more definitive way to declare your temporary storage. For example, you can describe your script's temporary storage variables using a stand-alone data area schema and associate it with your script.

Various step types involve referencing the script's data areas as well as support the ability to compare and move data to and from field elements residing in the data areas.

An Edit Data step supports the syntax to dynamically declare data areas as part of the step itself. This technique eliminates the need to statically declare a data area. Refer to Edit Data Syntax for more information on edit data commands and examples of the use of dynamic data areas.

> **Note**
>
> Some server based scripts may not use data areas as means of defining or exchanging data, depending on script type and the chosen scripting technique. Refer to The Big Picture Of Server Based Scripts for an overview of server scripts and their applicable scripting options.


Copyright © 2026, Oracle and/or its affiliates.

## Securing Script Execution

The system supports the ability to secure the execution of scripts by associating the script with an Application Service. Refer to The Big Picture of Application Security for more information. Application security is optional for user-invocable BPA scripts. If a script is not associated with an application service, all users may execute the script. Otherwise, only users that have **Execute** access to the application service may execute the script. For service scripts, the application service is required.

## Script Engine Version

Script engine version defines key information affecting the context and execution of the script.

- Script engine version values of 1, 2 and 3 define the version of the XML Path Language (XPath) to be used for the script. Versions 2 and 3 use the XPath 2 engine supplied by the XQuery team. This is the same engine used inside the Oracle database. The current script engine version 3 is a modified version that offers performance improvements without impacting existing version 2 scripts.

  The default script engine version is 3.0 for plug-in and service scripts. The default version for BPA scripts is 1.0 as higher level versions are not applicable.

  There are some additional details to note about script engine version 1.0:

  – The XPath library used is Jaxen

  – For BPA scripts, it uses the browser's xpath and XML support except for Internet Explorer where the XSXML parser is used.

  – Xpath 1 (and even JavaScript) uses floating point arithmetic, which means that adding a collection of numbers with two decimal places might end up with a value of 10779.079999999998 instead of 10779.08

- A Script Engine Version value of **Groovy** is only applicable to plug-in scripts and Groovy Library scripts. It is used to indicate that only Groovy related step types are used in the script and signals to the system that there is no need to convert the data to and from an XML interface. This allows for greater efficiency in script execution. This is the default value for Groovy Library scripts and cannot be changed.
- The value **Framework Version 2.1 Compatibility Mode** remains for upgrade purposes. This value should only be applicable to early versions of BPA scripts using syntax that is incompatible with XPath syntax.

# The Big Picture Of BPA Scripts

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Scripts to better understand the basic concept of scripts.

Users may require instructions in order to perform certain tasks. You may set up scripts that step a user through your business processes. For example, you might want to create scripts to help users do the following:

- Add a new person to an existing account
- Set up a customer to pay automatically


Copyright © 2026, Oracle and/or its affiliates.

- Modify a customer who no longer wants to receive marketing information
- Modify a customer's web password
- Record a trouble order
- Merge two accounts into one account
- Fix a bill with an invalid rate

Users execute these business processes by initiating BPA scripts (BPA). Users can also define their favorite BPA scripts in their user preferences.

BPA scripts may be designed by either of the following approaches:

- **Self-contained** and standalone flows. With this approach, the script is executed as an independent flow that cannot interact with the current application pages. Every step in these flows is contained within the script itself, including prompting for additional information from the user and performing server functions etc.
- **Guiding** flows that surf on top of one or more application pages and interact with these pages in parallel to script execution.

> ⚠ **Caution**
>
> This approach is considered legacy functionality and as such limited to fixed pages. It does not work with portal pages. With portals being the product's modern user interface technology, the overall support for this type of BPA scripts is planned to be deprecated in a future release.

> ⚠ **Caution**
>
> Future upgrade issues. There may be times when fixed pages are changed or converted to portals to leverage modern user experience capabilities. Such changes made by the base-package will necessitate a redesign of these guided flows. Please refer to the release notes for such changes.

The topics in this section describe background topics relevant to BPA scripts.

- How To Invoke Scripts
- Developing and Debugging Your BPA Scripts
- Launching A Script From A Menu
- Launching A Script When Starting The System
- Executing A Script When A To Do Entry Is Selected
- The Big Picture Of Script Eligibility Rules

## How To Invoke Scripts

Refer to Initiating Scripts for a description of how end-users initiate scripts.

The **Accessibility Option** setting for a BPA script may be used to control how it can be initiated. Set this value to **Accessible from Script Menu** for any script that may be launched as a stand-alone script. Scripts with this configuration may be linked to a navigation option so that they may be invoked from a menu and may be configured by a user as a favorite script. Set this value to **Not Accessible from Script Menu** for any script that cannot be launched on


Copyright © 2026, Oracle and/or its affiliates.

its own. For example, any script that is a invoked as a sub-script from another script should have this setting. In addition, any script that is designed to be launched from within a specific portal where certain data is provided to the script should include this setting.

## Developing and Debugging Your BPA Scripts

We recommend considering the approaches outlined below when you construct scripts.

While designing your scripts, determine the most maintainable way to set them up. Rather than creating complex, monolithic scripts, we recommend dividing scripts into smaller sections. For example

- Determine if several scripts have similar steps. If so, set up a script that contains these common steps and invoke it from the main scripts using a **Perform script** step.
- Determine if a large script can be divided into logical sections. If so, set up a small script for each section and create a "master script" to invoke each sub script via a **Transfer control** step.

For debugging purposes, you may find it helpful to categorize the step types into two groups: those that involve some type of activity in the script area, and those that don't. The following step types cause activity in the script area: **Height**, **Display text**, **Prompt user**, **Input data**, **Input Map**, **Set focus to a field**.

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

The rest of the step types are procedural and involve no user interaction. There are two techniques you can use to assist in debugging these step types.

- You can instruct the system to display text in the script area.
- You can display an entire data area (or a portion thereof) in the script area by entering **%+...+%** where **...** is the name of the node whose element(s) should be displayed.

> ⓘ **Note**
>
> **Time saver.** When you develop a new BPA script, change your user preferences to include the script as your first "favorite". This way, you can press `Ctrl+Shift+1` to invoke the script (eliminating the need to select it from the script menu).

## Launching A Script From A Menu

You can create menu items that launch BPA scripts rather than open a page. To do this, create a navigation option that references your script and then add a menu item that references the navigation option.

If the navigation option is referenced on a context menu and the navigation option has a "context field", a temporary storage variable will be created and populated with the unique identifier of the object in context. For example, if you add a "script" navigation option to the bill context menu and this navigation option has a context field of BILL_ID, the system will create a


Copyright © 2026, Oracle and/or its affiliates.

temporary storage variable called BILL_ID and populate it with the respective bill id when the menu item is selected.

## Launching A Script When Starting The System

You can set the system to launch a script upon startup. This may be used for integrations from an external system or for use cases within the application where a link has been configured to launch the application in a new browser tab, allowing the original page to remain open. Refer to Creating Application Links in External Applications for details of what is supported.

## Executing A Script When A To Do Entry Is Selected

The system creates To Do entries to highlight tasks that require attention (e.g., records in error). Users can complete many of these tasks without assistance. However, you can set up the system to automatically launch a script when a user selects a To Do entry. For example, consider a To Do entry that highlights a bill that's in error due to an invalid mailing address. You can set up the system to execute a script when this To Do entry is selected by a user. This script might prompt the user to first correct the customer's default mailing address and then re-complete the bill.

The following points provide background information to help you understand how to implement this functionality:

- Every To Do entry references a To Do type and a message category and number. The To Do type defines the category of the task (e.g., bill errors). The message number defines the specific issue (e.g., a valid address can't be found.). Refer to The Big Picture of System Messages for more information about message categories and numbers.
- When a user drills down on a To Do entry, either a script launches OR the user is transferred to the transaction associated with the entry's To Do type. You control what happens by configuring the To Do type accordingly:
  - If you want to launch a script when a user drills down on an entry, you link the script to the To Do type and message number. Keep in mind that you can define a different script for every message (and some To Do types have many different messages).
  - If the system doesn't find a script for an entry's To Do type and message number, it transfers the user to the To Do type's default transaction.

> **Note**
>
> **How do you find the message numbers?** We do not supply documentation of every To Do type's message numbers (this list would be impossible to maintain and therefore untrustworthy). The best way to determine which message numbers warrant a script is during pre-production when you're testing the system. During this period, To Do entries will be generated. For those entries that warrant a script, simply display the entry on To Do maintenance. On this page, you will find the entry's message number adjacent to the description.

- These types of scripts invariably need to access data that resides on the selected To Do entry. Refer to How To Use To Do Fields for the details.

## The Big Picture Of Script Eligibility Rules

You can configure eligibility criteria on the scripts to limit the scripts that a user sees in the script search. For example, you could indicate a script should only appear on the script menu if


Copyright © 2026, Oracle and/or its affiliates.

the user belongs to the level 1 customer service representative group. You may also indicate that a script should only appear if the data a user is viewing has certain criteria. For example, if you are using Oracle Utilities Customer Care and Billing, you can indicate that a script should only appear if the current account's customer class is residential. By doing this, you avoid presenting the user with scripts that aren't applicable to the current data in context or the user's role.

The topics in this section describe eligibility rules.

- Script Eligibility Rules Are Not Strictly Enforced
- You Can Mark A Script As Always Eligible
- You Can Mark A Script As Never Eligible
- Criteria Groups versus Eligibility Criteria
- Defining Logical Criteria
- Examples Of Script Eligibility Rules

## Script Eligibility Rules Are Not Strictly Enforced

The script search gives a user a choice of seeing all scripts or only scripts that are eligible (given the current data in context and their user profile). This means that it's possible for a script that isn't eligible for the given context data / user to be executed via this search. In other words, the system does not strictly enforce a script's eligibility rules.

It might be more helpful to think of eligibility rules as "highlight conditions". These "highlight conditions" simply control whether the script appears in the script search when a user indicates they only want to see eligible scripts.

## You Can Mark A Script As Always Eligible

If you don't want to configure eligibility rules, you don't have to. Simply indicate that the script is always eligible.

## You Can Mark A Script As Never Eligible

If you have scripts that you do not want a user to select from the script menu, indicate that it is never eligible. An example of a script that you wouldn't want a user to select from the menu is one that is launched when a To Do entry is selected. These types of scripts most likely rely on data linked to the selected To Do entry. As a result, a user should only launch scripts of this type from the To Do entry and not from the script menu.

## Criteria Groups versus Eligibility Criteria

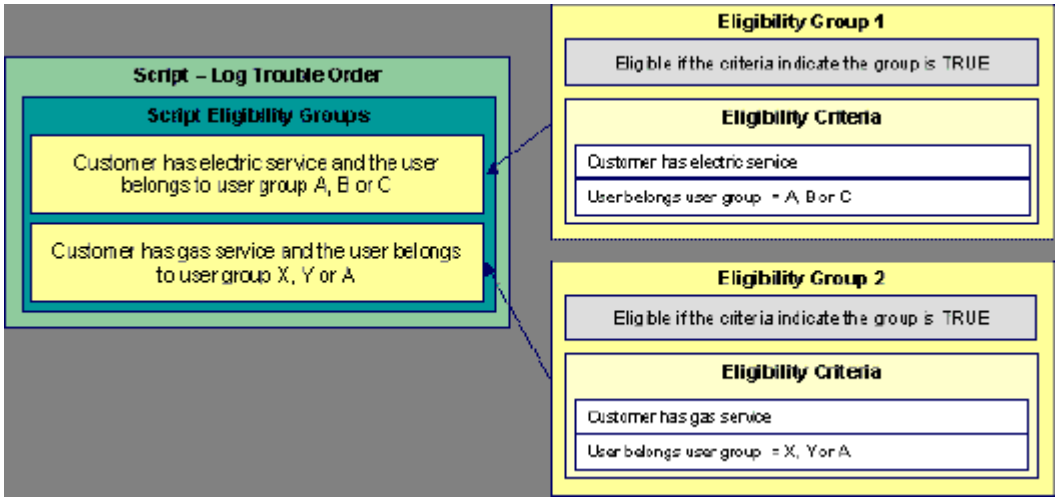
Before we provide concrete examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A script's criteria groups control whether a user is eligible to choose a script. At a high level, it works like this:

- A criteria group has one or more eligibility criteria. A group's criteria control whether the group is considered true or false.
- When you create a group, you define what should happen if the group is true or false. You have the following choices:
  - The user is eligible to choose the script.
  - The user is not eligible to choose the script.
  - The next group should be checked.


Copyright © 2026, Oracle and/or its affiliates.

We will use the following example from Oracle Utilities Customer Care and Billing to help illustrate these points. Assume a script is only eligible if:

- The customer has electric service and the user belongs to user group A, B or C.
- OR, the customer has gas service and the user belongs to user group X, Y or A.




This script requires two eligibility groups because it has two distinct conditions:

- IF (Customer has electric service AND (User belongs to user group A, B or C))
- IF (Customer has gas service AND (User belongs to user group X, Y or A))

If either condition is true, the script is eligible.

You would need to set the following criteria groups in order to support this requirement:

| Group No. | Group Description | If Group is True | If Group is False |
|---|---|---|---|
| 1 | Customer has electric service and the user belongs to user group A, B or C | **Eligible** | **Check next group** |
| 2 | Customer has gas service and the user belongs to user group X, Y or A | **Eligible** | **Ineligible** |

The following criteria are required for each of the above groups:

| Group 1: Customer has electric service and the user belongs to user group A, B or C | | | | |
|---|---|---|---|---|
| Seq | Logical Criteria | If Eligibility Criteria is True | If Eligibility Criteria is False | If Insufficient Data |
| 10 | Customer has electric service | **Check next condition** | **Group is false** | **Group is false** |
| 20 | User belongs to user group A, B or C | **Group is true** | **Group is false** | **Group is false** |

| Group 2: Customer has gas service and the user belongs to user group X, Y or A | | | | |
|---|---|---|---|---|
| Seq | Logical Criteria | If Eligibility Criteria is True | If Eligibility Criteria is False | If Insufficient Data |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | |
|---|---|---|---|---|
| 10 | Customer has gas service | **Check next condition** | **Group is false** | **Group is false** |
| 20 | User belongs to user group X, Y or A | **Group is true** | **Group is false** | **Group is false** |

The next section describes how you might configure the specific logical criteria in each of the groups.

## Defining Logical Criteria

When you set up an eligibility criterion, you must define two things:

- The field to be compared
- The comparison method

You have the following choices in respect of identifying the *field to be compared* :

- You can execute an algorithm to retrieve a field value from somewhere else in the system.
- Some products may support choosing a characteristic linked to an appropriate object in the system (such as an account or person).

You have the following choices in respect of identifying the *comparison method*:

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value.
- You can execute an algorithm that performs the comparison (and returns True, False or Insufficient Data). This is also a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

The Examples Of Script Eligibility Rules provide examples to help you understand this design.

## Examples Of Script Eligibility Rules

The topics in this section provide examples about how to set up script eligibility rules.

- A Script With A Time Span Comparison
- A Script With Service Type Comparison

## A Script With A Time Span Comparison

A script that is only eligible for senior citizens has the following eligibility rules:

- Customer class = Residential
- Birth date equates to that of a senior citizen

These rules require only one eligibility group on the script. It would look as follows:

| Group No. | Group Description | If Group is True | If Group is False |
|---|---|---|---|
| 1 | Residential and Senior Citizen | **Eligible** | **Ineligible** |

The following criteria will be required for this group:

| |
|---|
| Group 1: Residential, Calif, Senior |


Copyright © 2026, Oracle and/or its affiliates.

| Seq | Field to Compare | Comparison Method | If True | If False | If Insufficient Data |
|---|---|---|---|---|---|
| 10 | Algorithm: retrieve account's customer class | = R | **Check next condition** | **Group is false** | **Group is false** |
| 30 | Person characteristic: Date of Birth | Algorithm: True if senior | **Group is true** | **Group is false** | **Group is false** |

The first criterion is easy; it calls an algorithm that retrieves a field on the current account. This value, in turn, is compared to a given value. If the comparison results in a True value, the next condition is checked. If the comparison doesn't result in a True value, the **Group is false** (and, the group indicates that if the group is false, the script isn't eligible).

The last criterion contains a time span comparison. Time span comparisons are used to compare a date to something. In our example, we have to determine the age of the customer based on their birth date. If the resultant age is > 65, they are considered a senior citizen. To pull this off, you can take advantage of a comparison algorithm supplied with the base script as described below.

- Field to Compare. The person characteristic in which the customer's birth date is held is selected.
- Comparison Method. We chose a comparison algorithm that returns a value of **True** if the related field value (the customer's date of birth) is greater than 65 years (refer to `SECC-TIMESPN` for an example of this type of algorithm). Note that this time span algorithm can be used to compare days, weeks and months as well.

You'll notice that if a value of **True** is returned by the **True if senior** algorithm, the group is true (and we've set up the group to indicate a true group means the script is eligible).

## A Script With Service Type Comparison

Imagine a script that is only eligible if the current customer has gas service and the user belongs to user groups A, B or C. This script would need the following eligibility rules:

- Customer has gas service
- User belongs to user group A, B, or C

These rules require only one eligibility group on the script. It would looks as follows:

| Group No. | Group Description | If Group is True | If Group is False |
|---|---|---|---|
| 1 | Has gas service and user is part of user group A, B or C | **Eligible** | **Ineligible** |

The following criteria are required for this group:

| Group 1: Has gas service and user is part of user group A, B, or C | | | | | |
|---|---|---|---|---|---|
| Seq | Field to Compare | Comparison Method | If True | If False | If Insufficient Data |
| 10 | Algorithm: check if customer has gas service | = True | **Check next condition** | **Group is false** | **Group is false** |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | | |
|---|---|---|---|---|---|
| 20 | Algorithm: check if user belongs to user group A, B or C | = True | **Group is true** | **Group is false** | **Group is false** |

Both criteria are similar - they call an algorithm that performs a logical comparison. These algorithms are a bit counter intuitive (but understanding them provides you with another way to implement complex eligibility criteria):

The first criterion works as follows:

- Field to Compare. We chose a "field to compare" algorithm that checks if the current account has service agreements that belong to a given set of service types. It returns a value of **True** if the customer has an active service agreement that matches one of the service types in the algorithm. In our example, the "check if customer has gas service" algorithm returns a value of **True** if the customer has at least one active service agreement whose SA type references the gas service type. The "check if customer has electric service" algorithm is almost identical, only the service type differs.
- Comparison Method. We simply compare the value returned by the algorithm to True and indicate the appropriate response.

The second criterion works similarly:

- Field to Compare. We chose a "field to compare" algorithm that checks if the user belongs to any user group in a set of user groups. It returns a value of **True** if the user belongs to at least one user group defined in parameters of the algorithm. Refer to algorithm type `SECF-USRNGRP` for an example of this type of logic.
- Comparison Method. We simply compare the value returned by the algorithm to True and indicate the appropriate response.

> **Note**
>
> **Bottom line.** The "field to compare" algorithm isn't actually returning a specific field's value. Rather, it's returning a value of **True** or **False**. This value is in turn, compared by the "comparison method" and the group is set to true, false or check next accordingly.

# The Big Picture Of Server-Based Scripts

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Scripts to better understand the basic concept of scripts.

Server-based scripts allow an implementation to configure backend business processes. The system supports three types of server-based scripts, `Plug-In` scripts, **Service** scripts and **Groovy Library** scripts.

- Plug-in scripts allow an implementation to develop routines that are executed from the system's various plug-in spots. For example, an implementation could configure a plug-in script that is executed every time an adjustment of a given type is frozen.


Copyright © 2026, Oracle and/or its affiliates.

- Service scripts allow an implementation to develop common routines that are invoked from both front-end and back-end services. For example, an implementation could create a service script that terminates an account's automatic payment preferences. This service script could be invoked from a BPA script initiated by an end-user when a customer asks to stop paying automatically, and it could also be executed from a plug-in script if a customer fails to pay their debt on time. Service scripts are typically written using xpath scripting.
- Groovy Library scripts allow an implementation to develop groups of common routines written in the Groovy language which may be invoked from **Groovy Member** step types.

The topics in this section describe background topics relevant to server-based scripts.


- Additional Coding Options For Server Scripts
- Plug-In Scripts
- Service Scripts
- Groovy Library Scripts
- Debugging Server-Based Scripts


# Additional Coding Options For Server Scripts

Server based scripts often perform complex functions best supported by coding in languages with more comprehensive command sets than the base script steps. The system supports two common third-party languages for this purpose.

**XML Path Language** (XPath) is a language for querying and evaluating elements or nodes in an XML document. XPath commands and expressions can be used directly within edit data step types. The script engine version is used to define the applicable XPath version.

**Groovy** is an object-oriented, dynamic language for the Java platform. The framework supports the use of Groovy within server-based scripts to provide restricted and controlled access to Java-like code, particularly for cloud based implementations. The following topic provides more information on how to incorporate Groovy code into scripts.


- Using Groovy Within Scripts


## Using Groovy Within Scripts

Groovy code can be incorporated in scripts using the step type **Groovy Members**. For each script with Groovy code, there will be a single Groovy class created by concatenating all **Groovy Members** steps.

For security, the product and third party Java classes available for scripting in Groovy are restricted. The allowable base classes may be viewed in the Groovy Javadocs viewer via the 'Java Docs' action in the Script portal. The list of allowable third party classes can be viewed via the 'Third Party Groovy Allowlist' action also in the Script portal.

> **Note**
>
> This system supports the use of Groovy for back end processing purposes. It is not intended for user interfaces. Groovy is also not applicable to BPA scripts.

### Groovy Script Engine

For scripts invoked from within Java and entirely written in Groovy should use a script engine version of **Groovy**. The system processes this type of script in a similar way to code written in



Copyright © 2026, Oracle and/or its affiliates.

Java. This avoids the need to convert the script to and from XML format and allows the use of code that acts directly on the system objects with consequent performance benefits.

The following script types support the Groovy engine version:

**Plug In Scripts**

Plug-in Scripts can be configured to use the Groovy engine if they contain only Groovy oriented step types. The system provides an automatically generated superclass that defines the plug-in spot API. Internally, the Groovy code must conform to the system conventions for Java based algorithm types, including the inclusion of an 'invoke' method that is the plug-in entry point, and the definition of 'soft' parameters using annotations.

**Groovy Library Scripts**

Groovy Library Scripts provide the ability to create groups of common routines written in Groovy that can be called from within other scripts. Scripts of this type must include a single step type of **Groovy Library Interface** in which the publicly available methods in the library are listed. The supporting code for those methods is defined in one or more **Groovy Members** step types within the library script. The methods defined in the library can accept arguments and return values of any type. Scripts of this type use the Groovy engine by default and cannot include scripting step types.

Scripts that need to invoke methods from a Groovy library can use the createLibraryScript method provided by the system to instantiate the library interface. Ensure that the class being instantiated includes the .class filename extension.

### Invoking Groovy within the Scripting Engine

If a plug-in script or a service script is configured to use a scripting engine version, it can include a mixture of regular and **Groovy Members** step types. The script step types will define the process to be executed. The **Groovy Members** steps contain code that can be called from **Edit Data** step types within the script using the **invokeGroovy** command. Only Groovy methods that receive no arguments and return void are supported using this command. Refer to the section on edit data steps for more details.

For scripts using this option, the framework provides a superclass containing methods that support common scripting actions such as move commands, string evaluation, and methods to invoke business objects, business services and service scripts. Refer to the Groovy specific Javadocs for details of the supported methods.

### Importing Groovy Classes

All the Groovy classes used by your script should be explicitly declared in a **Groovy Imports** step type. When saving a script, if the system finds classes referenced in the script that are not explicitly declared, the system issues a warning and automatically adds import statements for the classes. Script writers should review the imports to confirm that they are the correct classes (in case there is a class with the same name in two different packages).

**Backward Compatibility**

In previous releases, the system did not require explicit imports and had logic to automatically import all callable classes available to Groovy at run-time. For any existing script that does not explicitly declare its imports, the system continues to fall back on importing all callable classes at runtime. However, the recommendation is to identify and update any such script as this is not the recommended practice.

The system provides a batch program `F1-CAGVY` that reports all scripts that required the automatic import step to successfully compile. In addition, this batch program is useful when the version of Groovy is upgrade to proactively detect possible issues in compilation of existing


Copyright © 2026, Oracle and/or its affiliates.

scripts that may occur due to changes in Groovy. Implementations are encouraged to run this batch program in the short term to identify and fix any scripts missing explicit imports and also for any Groovy version upgrade to review any implementation owned scripts that may be reported.

### Complex SQL Statements in Groovy

> **Important**
>
> Whenever you are designing an SQL statement, it is strongly recommended to work with your DBA to ensure the query meets performance standards.

There is a string size limitation in Groovy of 65,535 when using createPreparedStatement. This section highlights valid syntax to use to get around this limitation, without the risk of SQL injection.

#### Technique 1 - Simple String Concatenation with '+'

The createPreparedStatement method supports concatenating multiple long strings using the + (plus) operator as a connector.

```
def query = createPreparedStatement("SELECT * FROM TABLE_NAME"
+ "WHERE FIELD_A = 'VALUE1'"
+ "AND FIELD_B = 'VALUE2'"), 'My Query');
```

#### Technique 2 - Special Method in Prepared Statement

The system provides a method concatenateGroovyStrings which accepts varargs of strings:

```
concatenateGroovyStrings(String... args)
```

This method may be used directly in createPreparedStatement as an argument:

```
def query = createPreparedStatement(concatenateGroovyStrings("SELECT* ",
"FROM TABLE_NAME", "My query");
```

#### Technique 3 - Reference a Variable in Prepared Statement

The system supports defining a variable as the first argument in createPreparedStatement as long as the variable was built using the concatenateGroovyStrings method.

```
def concatStr = concatenateGroovyStrings("SELECT * ", "FROM ", "TABLE_NAME");
def query = createPreparedStatement(concatStr, "My Query");
```

The system validates that the variable used as an argument was created using the concatenateGroovyStrings method. The system also supports a straight reassignment of the variable.

```
def concatStr = concatenateGroovyStrings("SELECT * ", "FROM ", "TABLE_NAME");
def newStr = concatStr;
def query = createPreparedStatement(newStr, "My Query");
```


Copyright © 2026, Oracle and/or its affiliates.

Following on with this one, the prepared statement can combine strings and variables, as long as the variable was created using the concatenateGroovyStrings method.

```
if (restrictByField()){
def fieldClause = concatenateGroovyStrings(" AND FIELD_NAME = 'VALUE'");
}
def query = createPreparedStatement("SELECT * FROM TABLE_NAME WHERE OWNER_FLG
= 'CM'" + fieldClause, "My Query");
```

### Handling Elements in an Array

When attempting to build an SQL statement, you may have use cases that require looping through a list and dynamically adding content to the SQL statement from within the list. Additional helper methods are provided for these more complex use cases.

**Use Case 1:** Include one or more values in an "in" clause.

If you have a dynamic list of values that you want to include in an "in" clause, the following helper method handles generating the bind variables.

```
concatenateSqlInClause(String columnExpr, String bindName, int start, int
count)
```

The columnExpr and bindName should be string literals. Example: you have a list of To Do Types and you want to include that list in an "in" clause.

```
def sql = concatenateGroovyStrings("SELECT TD_ENTRY_ID FROM CI_TD_ENTRY",
                   "WHERE ENTRY_STATUS_FLG <> 'C'",
                   "AND",
         concatenateSqlInClause("TD_TYPE_CD", ":tdType", 1,
tdTypeInputCount),"
         )
```

The result of the above would be the following if there are 5 To Do Types in the list:

```
SELECT TD_ENTRY_ID FROM CI_TD_ENTRY WHERE ENTRY_STATUS_FLG <> 'C'
                   AND TD_TYPE_CD IN
(:tdType1,:tdType2,:tdType3,:tdType4,:tdType5)
```

**Use Case 2:** Create a Bind Name with an index

The following helper method allows you to create bind names for use in creating an SQL statement:

```
concatenateBindName(String bindName, int index)
```

For example, within a 'for' loop where 'i' is the index name,

```
concatenateBindName(:tdType, i)
```

The results of this method can be added to the SQL statement being prepared using concatenateGroovyStrings.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Use the Script query > Step Information search option to look for base scripts that use the helper method to find examples of this technique.

**Use Case 3:** Using the index value

There may be use cases where you need the literal value of an index variable. For this use case, the following method is available:

```
literalFor(int i)
literalFor(long l)
```

This may be used to dynamically build the column names, for example when you want to search for records by the prime key value and the number of prime keys is variable. Example:

```
String sql = concatenateGroovyStrings("SELECT * FROM F1_MO_UPD WHERE ")
for (int i = 1; i <= pkCount; i++) {
    sql = concatenateGroovyStrings(sql, "PK_VALUE", literalFor(i), " = ",
concatenateBindName(":pkVal", i))
    if (i < pkCount) {
        sql = concatenateGroovyStrings(sql, " AND ")
    }
```

For an object that has a two-part primary key, the above result is as follows:

```
SELECT * FROM F1_MO_UPD WHERE PK_VALUE1 = :pkVal1
    AND PK_VALUE2 = :pkVal2
```

# Plug-In Scripts

> **Note**
>
> This section assumes you are familiar with the notion of plug-in spots (algorithm entities) and plug-ins. See The Big Picture Of Algorithms for more information.

As an alternative to writing a java program for a plug-in spot, the framework supports creating plug-ins using the defined script steps, Xpath commands, Groovy code or a combination of these three options.

The following topics describe basic concepts related to plug-in scripts.

- A Plug-In Script's API
- Setting Up Plug-In Scripts

## A Plug-In Script's API

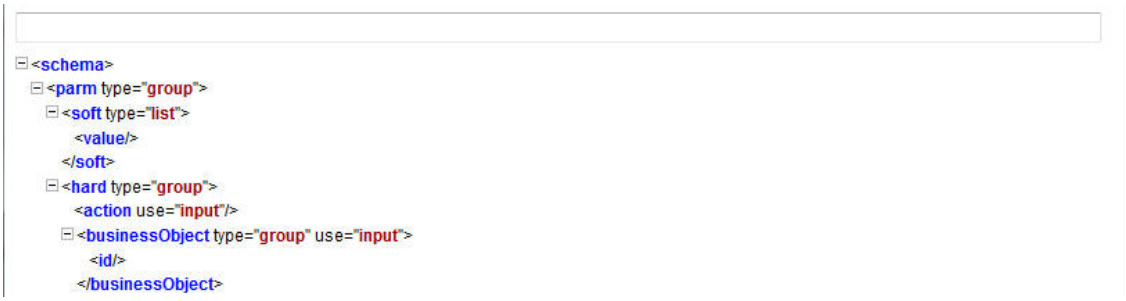
Like program-based plug-ins, plug-in scripts:

- Run on the application server.


Copyright © 2026, Oracle and/or its affiliates.

- Have their API (input / output interface) defined by the plug-in spot (i.e., plug-in scripts do not get to declare their own API).
- Can only be invoked by the "plug-in spot driver".

For plug-ins configured to use a script engine version, the best way to understand the script's API is to use the **View Script Schema** hyperlink to view its parameters data area schema.




Notice the two groups: soft and hard. If you are familiar with plug-in spots, you'll recognize these as the classic soft and hard parameters:

- The **soft** parameters are the values of the parameters defined on the algorithm. Notice they are not named - if you want to reference them in your plug-in script, you must do it by position.
- The **hard** parameters are controlled by the plug-in spot (i.e., the algorithm entity). Notice that this plug-in spot has a single input parameter called " **businessObject/id"**. Also notice the `use=` attribute - this shows that this parameter is input-only (i.e., you can't change it in the plug-in script).

> **Note**
>
> **XPath.** You can click on an element name to see the XPath used to reference the element in your script.

Plug-ins configured to use the **Groovy** engine version do not use an XML interface for the API and instead are processed in the same way as Java algorithms. The framework supplies a dynamically generated superclass that implements the plug-in spot for Groovy objects. Use the **View Script Superclass** hyperlink to view this superclass and the methods to set and get the hard parameters.

> **Note**
>
> For plug-in scripts using the **Groovy** engine version, soft parameters do not appear in the plug-in spot API as defined by the superclass. Plug-ins using only Groovy code define their soft parameters using annotations, in a similar way to Java algorithms, and fetch those values using methods defined in the algorithm code.

## Setting Up Plug-In Scripts

The following points describe how to implement a plug-in script:

- Compose your plug-in script, associating it with the appropriate algorithm entity (plug-in spot).


Copyright © 2026, Oracle and/or its affiliates.

- Create a new algorithm type for the respective algorithm entity, referencing your plug-in script as the program to carry out the algorithm type's function. Only plug-in scripts associated with the algorithm entity may be referenced on the algorithm type.
- Set up an algorithm for the respective algorithm type and plug it in where applicable. Refer to Setting Up Algorithm Types for more information.

# Service Scripts

BPA scripts run on the client's browser and guide the end-users through business processes. Service scripts run on the application server and perform server-based processing for BPA scripts, zones and more. You may want to think of a service script as a common routine that is set up via scripting (rather than programming).

The following topics describe basic concepts related to service scripts.

- A Service Script's API
- Invoking Service Scripts

## A Service Script's API

As with any common routine, a service script must declare its input / output parameters (i.e., its API). A service script's API is defined on its schema.

> **Note**
>
> Refer to Schema Nodes and Attributes for a complete list of the XML nodes and attributes available to you when you construct a schema.

## Invoking Service Scripts

Any type of script configured to use a scripting engine version may invoke a service script:

- A BPA script may invoke a service script to perform server-based processing.
- Plug-in scripts may invoke a service script (like a "common routine").
- A service script may call another service script (like a "common routine").

Map zones may be configured to invoke service scripts to obtain the data to be displayed. Refer to Map Zones for more information.

Inbound web services support interaction with service scripts allowing the outside world to interact directly with a service script.

You can also invoke a service script from a Java class.

# Groovy Library Scripts

Just as service scripts can define common routines written in scripting language, Groovy library scripts are used to define groups of common components or methods written in Groovy. Groovy library code runs on the application server and performs server-based processing for scripts that utilize Groovy code.

The following topics describe basic concepts related to Groovy library scripts.

- A Groovy Library Script's API


Copyright © 2026, Oracle and/or its affiliates.

- Invoking Groovy Library Methods

### A Groovy Library Script's API

A Groovy library script's API is composed of one or more public methods whose code is defined in the script's steps. Those methods are defined in a step type of **Groovy Library Interface**. A Groovy library script must have one (and only one) step of this type.

### Invoking Groovy Library Methods

Any type of script that supports the **Groovy Members** step type may invoke common methods defined in Groovy library scripts.

Code within a **Groovy Members** step must create an instance of the Groovy library interface definition to enable the interface methods to be invoked. Refer to the topic Using Groovy Within Scripts for more information.

## Debugging Server-Based Scripts

The server can create log entries to help you debug your server scripts.

The logs contain a great deal of information including the contents of the referenced data area for **Move data**, **Invoke business object**, **Invoke business service** and **Invoke service script** steps.

Refer to the Debug Mode topic in the Configuration Tools chapter for details of how to execute the application in debug mode.

# Defining Scripts

The script maintenance portal is used to maintain your scripts.

You can access the portal from the **Admin Menu**, then **System**, then **Script**. You are brought to a query portal with options for searching for a specific script. Once a script has been selected you are brought to the maintenance portal to view and maintain the selected record.

Scripts may be of the following types: **BPA Script**, **Plug-In Script**, **Groovy Library Script** and **Service Script**. Refer to The Big Picture Of BPA Scripts and The Big Picture Of Server Based Scripts for more information.

> ⚠ **Caution**
>
> Important! If you introduce a new script, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Script.** This zone provides information about the selected script. Refer to Securing Script Execution for more information.
- **Steps.** Use this zone to review and maintain the script's steps. You may update a single step or maintain all steps as a list. Refer to How To Set Up Each Step Type for an overview of the step types.

The following zones may appear as part of the portal's **Schema** tab page. This tab page only appears for service scripts and BPA Scripts.


Copyright © 2026, Oracle and/or its affiliates.

- **Schema Editor.** This zone allows you to edit the script's schema using the Schema Designer tool. The purpose of the schema is to describe the data elements passed to and from the script.

> **Note**
>
> Refer to Schema Syntax and UI Hint syntax for a complete list of the XML nodes and attributes available to you when you construct a schema.

> **Tip**
>
> Click the **View Schema** hyperlink to view the business object's expanded schema definition. Doing this opens the schema viewer window.

The following zones may appear as part of the portal's **Data Areas** tab page. This tab page does not appear for Groovy Library scripts or plug-in scripts using the **Groovy** engine version.

- **Data Areas.** Use this zone to define the data areas used to pass information to and from the server or any other data area describing your temporary storage. Click on the **View Data Area** action to view the data area's schema in the schema viewer window.

The following zones may appear as part of the portal's **Eligibility** tab page. This tab page only appears for BPA scripts.

- **Script Eligibility.** Use this zone to define a BPA script's eligibility rules. Refer to the guidelines under The Big Picture Of Script Eligibility for more information on how to define this information.

The following zones may appear as part of the portal's **References** tab page.

- **Script References.** This zone list various configuration entities referencing the current script or its schema.

### Actions

The page supports the following special actions:

- **View Schema.** Click this link to view the script's expanded schema definition. Doing this opens the schema viewer window. This action is only available for BPA scripts and service scripts.
- **View Script As Text.** Click this link to view the internal scripting commands in a separate window.
- **Test.** Use this action to test your service script of BPA script.
  - For a service script, enter input values and save. The service is performed and output information is presented, but the system issues a roll-back so that the changes are not committed. Note that if your service uses the save-point dispatcher to perform any actions or if it issues a call to an external system, the roll-back does not apply to those actions. If the service issues an error the corresponding message is displayed. Note that the input and output testing user interface is dynamically rendered based on UIHints embedded in the service script's schema. As such, the user interface may be as elaborate or basic as the UIHints in that schema.
  - Testing a BPA script is only applicable to a BPA that does not rely on information passed to it from somewhere else. Because the system cannot detect if this is the


Copyright © 2026, Oracle and/or its affiliates.

case for your BPA, the Test button is always visible. Attempting to use it on a BPA that relies on information passed into it may show an error.

> **Note**
>
> This action is secured by a designated "Test a Service" application service and the "Generic Service Executor" application service and also requires security rights for creating a script, which is the alternative way for testing any service. It is recommended to allow access to this function in non-production environments only.

- **Java Docs.** Use this action to launch the Groovy Javadocs viewer.
- **Third Party Groovy Allowlist.** Use this action to review the allowlist of third party Groovy classes and verify the restricted list of classes available for Groovy coding in the script.
- **View Script Supreclass.** Use this action to view the code of the runtime generated superclass for the related plug-in spot's implementation.. This action appears only for plug-in scripts using an engine version of **Groovy**.
- How To Set Up Each Step Type
- Additional Topics

## How To Set Up Each Step Type

The contents of this section describe how to set up each type of step. Note that some step types are only applicable for BPA scripts and some step types are only supported for server based scripts. In addition, some BPA script steps are considered legacy functionality and therefore only supported on fixed pages and not portals. Each section indicates whether there are any script type limitations.

- Conditional Branch Steps
- Display Text Steps
- Edit Data Steps
- Go To Steps
- Height Steps
- Input Data Steps
- Invoke Business Object Steps
- Invoke Business Service Steps
- Invoke Function Steps
- Invoke Map Steps
- Invoke Service Script Steps
- Groovy Imports Steps
- Groovy Library Interface Steps
- Groovy Member Steps
- Label Steps
- Mathematical Operation Steps
- Move Data Steps


Copyright © 2026, Oracle and/or its affiliates.

- Navigate To A Page Steps
- Perform Script Steps
- Press A Button Steps
- Prompt User Steps
- Set Focus To A Field Steps
- Terminate Steps
- Transfer Control Steps

## Conditional Branch Steps

**Conditional branch** steps allow you to conditionally jump to a different step based on logical criteria. For example, you could jump to a different step in a script if the customer is residential as opposed to commercial. In addition, several fields are required for **Conditional Branch** steps:

**Compare Field Type** and **Compare Field Name** define the first operand in the comparison. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Current To Do Information**. Use this field type when the field being compared resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**.
- **Data Area**. Use this field type when the field being compared is one that you put into one of the scripts data areas in an earlier step. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the field being compared resides on one of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Predefined Value**. Use this field type when the field being compared is a global variable.
- **Temporary Storage**. Use this field type when the field being compared is one that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the field being compared resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

**Condition** defines the comparison criteria:

- Use >, <, =, >=, <=, <> (not equal) to compare the field using standard logical operators. Enter the comparison value using the following fields.
- Use **In** to compare the first field to a list of values. Each value is separated by a comma. For example, if a field value must equal `1`, `3` or `9`, you would enter a comparison value of `1,3,9`.
- Use **Between** to compare the field to a range of values. For example, if a field value must be between `1` and `9`, you would enter a comparison value of `1,9`. Note, the comparison is inclusive of the low and high values.

**Comparison Field Type**, **Comparison Field Name** and **Comparison Value** define what you're comparing the first operand to. The following points describe each field type:


Copyright © 2026, Oracle and/or its affiliates.

- **Current To Do Information**. Use this field type when the comparison value resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**.
- **Data Area**. Use this field type when the comparison value resides in one of the scripts data areas. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the comparison value resides on one of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Predefined Value**. Use this field type when the field being compared is a constant value defined in the script. When this field type is used, use **Comparison Value** to define the constant value. Refer to How To Use Constants In Scripts for instructions on how to use constants.
- **Temporary Storage**. Use this field type when the comparison value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the comparison value resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

> **Note**
>
> **Conditional field types.** The field types **Current To Do Information**, **Page Data Model** and **User Interface Field** are only applicable to BPA scripts. Only the field types **Temporary Storage** and **Predefined Value** are applicable when the Condition is **In** or **Between**.

The above fields allow you to perform a comparison that results in a value of **TRUE** or **FALSE**. The remaining fields control the step to which control is passed given the value:

- **If True, Go to** defines the step that is executed if the comparison results in a **TRUE** value.
- **If False, Go to** defines the step that is executed if the comparison results in a **FALSE** value.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Numeric Comparison**. Comparison of two values may be numeric or textual (left-to-right). Numeric comparison takes place only when values on both side of the comparison are recognized as numeric by the system. Otherwise, textual comparison is used. Fields for **Current To Do Information**, **Data Area**, **Page Data Model**, and **User Interface Field** types are explicitly associated with a data type and therefore can be recognized as numeric or not. This is not the case for fields residing in **Temporary Storage** or those set as **Predefined Values**. A **Temporary Storage** field is considered numeric if it either holds a numeric value moved to it from an explicitly defined numeric value (see above) or it is a resultant field of mathematical operation. A **Predefined Value** field is considered numeric if the other field it is compared to is numeric. For example, if a numeric field is compared to a **Predefined Value** the latter is considered numeric as well resulting in numeric value comparison. However, if the two fields are defined as **Predefined Values** the system assumes their values are text strings and therefore applies textual comparison.

## Display Text Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

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

**Display text** steps cause a text string to be displayed in the script area. Steps of this type can be used to provide the user with guidance when manual actions are necessary. In addition, they can be used to provide confirmation of the completion of tasks.

The information you enter in the **Text** field is displayed in the script area when the step is executed.

The text string can contain substitution variables and HTML formatting commands. Also note that for debugging purposes, you can display an entire data area (or a portion thereof) by entering **%+...+%** where **...** is the name of the node whose element(s) should be displayed.

## Edit Data Steps

**Edit data** steps provide a free format region where you can specify commands to control your script processing.

Enter your scripting commands in the **Edit Data Text** field. Click the adjacent icon to open a window providing more space for defining the edit data step.

In general, the syntax available within edit data mimics the commands available within the explicit step types. However, there are a few commands that are available only within edit data. For example, the two structured commands: **For**, and **If**.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Not all BPA step types are supported using the edit data syntax. Refer to the details below for more information on edit data commands and examples.

The topics in this section provide detail of the syntax supported in the edit data step type.

### Comments

You can place comments into your script using the double slash notation `//` in the first two characters of the edit data step. Example:

```
//
// quit with error
//
if ("not(customer/securityEnabled)")
terminate with error (8000, 1001 %1="customer/id" %2='not allowed');
end-if;
```

### Temporary Variables

Temporary variables can be declared within all types of scripts. They should be referenced by a leading single dollar sign ('$'). However, temporary variables behave differently in the various script types:

- In BPA Scripts temporary variables remain persistent from one BPA script to another (refer to the Perform Script and Transfer Control statements), which means that you can use temporary variables to communicate between BPA scripts.
- Temporary variables cannot be passed from a BPA script to a service or plug-in script. Only data area elements can be passed between these types of scripts.
- Within service and plug-in scripts, temporary variables remain persistent only for the life of the particular script that declared the variable. This means that temporary variables cannot be passed between plug-in scripts and service scripts, only global variables, context variables and data area elements can be passed between these types of scripts.

#### Declaring / Initializing / Defaulting Temporary Variables

When using a temporary variable, it should be declared or initialized with an appropriate value before using it. A typical method for declaring a variable is to simply move data to it in a move statement, for example.

```
move "0" to $index;
```

> **NOT_SUPPORTED**
>
> Refer to Move to a Temporary Variable for more information on implicit declaration of a temporary variable within a move statement.

For BPA scripts, as mentioned above, temporary variables may be passed from one BPA script to another. As such, it is common to reference a temporary variable in a BPA that should have been initialized by a previous BPA. However, if there is any reason that a temporary variable did not get initialized by a previous BPA, a reference to it will cause an error. It is good practice,


Copyright © 2026, Oracle and/or its affiliates.

therefore, to use the **default** statement that will initialize temporary variables that are not already created / initialized.

- The following statement will initialize the temporary variable $InputAction – but only if the temporary variable has not yet been initialized:

  ```
  default $InputAction;
  ```

- The following statement will set the value of the temporary variable $InputAction to 'read' – but only if the variable has not yet been initialized:

  ```
  default $InputAction as 'read';
  ```

> **Note**
>
> Scripts should take care not to define variables using a reserved keyword. The following table lists the reserved keywords.

| Keyword |
|---|
| **add** |
| **as** |
| **asError** |
| **bpa** |
| **branch** |
| **data** |
| **declareBO** |
| **declareBS** |
| **declareDA** |
| **declareMap** |
| **declareSS** |
| **default** |
| **delete** |
| **edit** |
| **element** |
| **else** |
| **end-edit** |
| **end-for** |
| **end-if** |
| **error** |
| **escape** |
| **evaluate** |
| **fastAdd** |
| **fastUpdate** |
| **for** |
| **goto** |
| **if** |
| **in** |
| **invokeBO** |
| **invokeBS** |


Copyright © 2026, Oracle and/or its affiliates.

| Keyword |
| --- |
| **invokeMap** |
| **invokeSS** |
| **label** |
| **map** |
| **move** |
| **navigate** |
| **navigateAndReloadDashboard** |
| **null** |
| **page** |
| **performScript** |
| **popup** |
| **read** |
| **readWithoutVersion** |
| **replace** |
| **suppress** |
| **target** |
| **terminate** |
| **to** |
| **transferControl** |
| **update** |
| **using** |
| **warn** |
| **with** |

### Context Variables

Context variables are only available within service scripts. The context variable will be available for the duration of the service script and all that it invokes. Therefore, you can use a context variable within a service script to communicate information to a lower level service script or schema. They should be referenced by leading double dollar signs ('$$').

> **Note**
>
> Because context variables are available for lower level scripts, they may sometimes be referred to as global variables or global context variables. But they should not be confused with global variables.

### Declaring / Initializing / Defaulting Context Variables

When using a context variable, it should be declared or initialized with an appropriate value before using it. A typical method for declaring a variable is to simply move data to it in a move statement, for example.

```
move 'context variable' to $$contextVariable;
```


Copyright © 2026, Oracle and/or its affiliates.

**NOT_SUPPORTED**

Refer to Move using a Context Variable for more information.

### Move Statement

The **move** statement copies a source value to a target. The following table highlights various options supported in the move statement.


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move to Element<br>**move "xpath" to "xpath";** | Moves a "tacent" to this Bill exp "cat" hr to "er" pace. / fo |

> **Note**
>
> An XPath expression is surrounded by double quotes.


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Description |
| --- | --- |

rmattedValue";


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Move statement (with X-Path/Concatenate function). |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | erson/lastName)" to "parm/fullName"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Moves a tenant with XPatch subscriptions (Name, Description, Name, c |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | tion.) " to " person/firstName" ; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Moves a tenant with XPath substring/full name (Neanmen'c |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | ion)" to "person/lastName"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Moves a "stem" or "twiting" (Xpath/substring function". p |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | arm/year"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move to Element<br>**move 'literal' to "xpath";**<br><br>**Note**<br>A literal value is surrounded by single quotes. | Moves a token at using a literal string. account/ |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | preferences[type="mail"]/text"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move to Element<br>**move 'Boolean' to "xpath";** | Moves (a component with Boolean a>0) to the Boolean's storing. |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | true' to "account/hasDebitBalance"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | end-if; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Moving a expression, which has a Boolean. |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Note that the "filter" or "enable" example below is ignored. |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | <n/a><c><u><n><d><e></schema> move "boolean(a |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | count[balance>0]) " to "account/hasDe |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | bitBalance"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move to Group<br>**move "xpath" to "xpath";** | Moves "tacoelement/customer" to "group" then sorter. |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | This system attempts to find the equivalent of the element in the Syntax Description |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax | Description |
| --- | --- | --- |
| | mg statement. move "account/custInfo/*" | mes and the group/list names from the source schem |

| Statement | Example Description |
| --- | --- |
| | at the "parsons/chem" a. For lists and groups, the de |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | fault behavior is to move all elements within the s |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | source group/list to the target group/list (eveni |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | if they are not defined in the target group/list). Th |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | ebusinessserviceF1-MoveByNamemaybeusedform |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | or regranular control of the move (without having t |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | o define each individual move statement). |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move using a Temporary Variable | When moving to a temporary location (Temporary Variable), no |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | [illegible] |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | arsonCe;ildCount; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | When moving from a "Arcomatic Balance" to a "variable" pathway/ariab |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Least Variable "Sys.Variable" by double quotes. **move $variab** |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | le "to "xpath"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move using a Context Variable<br>**move "xpath" to $$variable;**<br>**move $$variable to “xpath”;** | Context variables, source or target, are not exten |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | cardinal hour/any double quotes. move from a c |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Description |
| --- | --- |
| | ontext variable.move $ $contextVari |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | able to "MarketMessage/sender"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move using a Dynamic Location<br>**move "xpath" \| 'literal' to evaluate("xpath" \| $variable);**<br>**move evaluate("xpath" \| $variable) to "xpath" \| $variable;** | The evaluate statement allows you to move a source (or |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Example Description |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | "evaluate(" |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Description |
| --- | --- |
| | schemaLocation") to $Variable;// mo |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Description |
| --- | --- |
| | ve<br><br>evaluate(<br><br>$Variable)<br><br>to<br><br>"schemaLoc |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | ation"; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Description |
| --- | --- |
| Move escape<br>**move escape("xpath" \| $variable) to "xpath" \| $variable;** | Move escape("xpath" \| $variable) to "xpath" \| $variable; |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Example Description |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example | Description |
| --- | --- | --- |
| | "schemaLocation". | Qursource text value for HTML content and escapes |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | it, i.e. replaces any HTML-like characters with |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Description |
| --- | --- |
| | special characters that are escaped from HTML rend |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | ering. By doing so the text would be displayed as pl |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | a inline text when displayed as part of an HTML element. |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| Move null<br>**move null to "xpath";** | Removes a node in a full or a fit scrub mode XML instance |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Document (chrsoulogh/ressecurity/Etnaxbolfmove')nulg' |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Note that you can specify either an outline or the |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | Use the expression, or a group name. Remove your subscription and |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | resource tag error tuple or error with a handle such as "Fix 'd*' element (with lobee |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Description |
| --- | --- |
| | limit (actual from prospecting. Enabled) ") mov |


Copyright © 2026, Oracle and/or its affiliates.

| Statement | Example Syntax Explanation |
| --- | --- |
| | e<br><br>n<br>u<br>l<br>l<br><br>t<br>o<br><br>"<br>c<br>u<br>s<br>t<br>o<br>m<br>e<br>r<br>/<br>*<br>"<br>;<br>e<br>n<br>d<br>-<br>i<br>f<br>; |

### Go To Statement

The edit data step supports functionality analogous to the Go To step type. The syntax is **goto label;** where the label represents another location within the edit data text field (identified by this label) or represents another step in the script.


Copyright © 2026, Oracle and/or its affiliates.

The following is an example of going to another location in the same step identified by the label **addSpouse**.

```
if ("string(parm/spouse/name) != $BLANK")
  goto addSpouse;
end-if;
addSpouse: invokeBO 'Person' using "parm/spouse" for add;
```

The following is an example of going to a different step within the same script. The step sequence is the reference used as the label.

```
if ("string(parm/spouse/name) != $BLANK")
  goto 110;
end-if;
.
.
.
110: invokeBO 'Person' using "parm/spouse" for add;
```

### Conditional Branch Statement

The edit data step supports functionality analogous to the Conditional Branch step type. The syntax is **branch (“xpath”) goto label else label;** where:

- The XPath condition in the **branch** statement must evaluate to a Boolean value of True or False.
- The targets for the **goto** and **else** statements are labels that represent another location within the edit data text field (identified by this label) or represent another step in the script.

The following example uses labels for **addSpouse** and **addAccount**

```
branch ("string(parm/spouse/name) != $BLANK") goto addSpouse else addAccount;
```

### If Statement

The **if** statement is similar to the conditional branch statement. Either can be used to structure the logic of your script. This statement may optionally include an **else** statement but it should always end with an `end-if` statement.

> **Note**
>
> This is an example of a statement that is not represented as a separate step type. It is only available within the edit data text.

The syntax is **if (“xpath”) else end-if;**. The XPath condition must evaluate to a Boolean value of True or False. The following are some examples.

Example where the XPath contains a simple logical condition.

```
if ("string(parm/spouse/name) != $BLANK")
  //
  // Create spouse since spouse name present
  goto addSpouse;
else
```


Copyright © 2026, Oracle and/or its affiliates.

```
  //
  // Create account without spouse
  goto addAccount;
end-if;
```

Example where the XPath contains a complex condition.

```
if ("string(parm/spouse/name) != $BLANK and string(parm/hasSpouse) = true or
boolean(parm/requireSpouse)")
  //
  // Create spouse since spouse name present
  goto addSpouse;
end-if;
```

Example of a stacked set of statements used to evaluate multiple possible values of a field.

```
if ("parm/rowCount = 0")
  //
  // no rows found
  goto quit;
end-if;
if ("parm/rowCount = 1")
  //
  // one row found
  goto process;
end-if;
if ("parm/rowCount > 1")
  //
  // more than one row found
  goto quit;
end-if;
quit: terminate;
```

The following XPath shows Boolean based on the existence of the node. In this example, if the node exists in the XML instance document being processed, the statement will evaluate to True. If no element is found, the statement evaluates to false.

> **Note**
>
> When treating XPath nodes as Boolean variables be aware that an empty node evaluates to True. Only a missing node return False.

```
if ("boolean(parm/spouse/name)")
  goto addSpouse;
else
  //
  // Create account without spouse
  goto addAccount;
end-if;

if ("not(parm/spouse/name)")
  //
```


Copyright © 2026, Oracle and/or its affiliates.

```
  // Create account without spouse
  goto addAccount;
else
  goto addSpouse;
end-if;
```

### For Statement

The **for** statement creates a list of nodes or values depending on your XPath expression. If you specify a list node then every child node of the list, along with its contents, will be available within the loop. If you specify a child node directly, then a list of values only will be available within the loop.

> **Note**
>
> For more information on creating new entries in a list, please refer to the creating a new list instance example.

> **Note**
>
> This is an example of a statement that is not represented as a separate step type. It is only available within the edit data text.

The syntax is **for ($variable in "xpathList") end-for;**. The XPath condition must evaluate to a Boolean value of True or False.

The following examples are based on this sample schema:

```
<schema>
  <SAList type="list">
    <id/>
    <balance/>
  </SAList>
  <SAContributor type="list">
    <id/>
  </SAContributor>
</schema>
```

Example that specifies the list node in the XPath expression where all child nodes are available for processing.

```
move "0" to $AccountBalance;
move "0" to $index;
for ($SAList in "SAList")
    move "$SAList/balance + $AccountBalance" to $AccountBalance;
    //
    // keep track of each SA contributing to the balance in the SA
Contributor list
    move "1 + $index" to $index;
    move "$SAList/id" to "SAContributor[$index]/id";
end-for;
```


Copyright © 2026, Oracle and/or its affiliates.

Example that specifies a child node within the list node in the XPath expression. Only values of that node are available for processing.

```
move "0" to $AccountBalance;
for ($SABalance in "SAList/balance")
    move "$SABalance + $AccountBalance" to $AccountBalance;
end-for;
```

Example that shows that a filter can be used to limit the rows selected by the **for** loop.

```
move "0" to $AccountDebitBalance;
for ($SAList in "SAList[Balance>0]")
    move "$SAList/balance + $AccountDebitBalance" to $AccountDebitBalance;
end-for;
```

Example that shows the use of a filter when specifying child nodes.

```
move "0" to $AccountCreditBalance;
for ($SABalance in "SAList[Balance<0]/balance")
    move "$SABalance + $AccountCreditBalance"
to $AccountCreditBalance;
end-for;
```

### List Processing

This section provides details about processing lists. The examples in this section reference the following schema:

```
<schema>
  <parm type="group">
    <name/>
  </parm>
  <Person type="group">
    <names type="list">
      <type/>
      <name/>
    </names>
  </Person>
</schema>
```

**Referencing a List Element.** You can move a value to a particular list instance by referencing an identifying node in the list within a filter. The syntax is **move "xpath" to "xpathList[filter]/element";** Example:

```
move "parm/name" to "Person/names[type='main']/name";
```

**Creating a New List Instance.** A special notation can be used within a move target statement to indicate a new list instance should be created. The "+" indicates to the script processor that a new instance of a list should be initiated for the target element. The syntax is **move "xpath" to "+xpathList";** Example:

```
move "parm/name" to "Person/+names/name";
```


Copyright © 2026, Oracle and/or its affiliates.

**Deleting a List Instance.** An XML list entry can be deleted from the database by moving an action attribute of 'delete' to the element name. To cause a database delete of a list entry requires an attribute of action="delete" in the target node and a subsequent update BO interaction. The syntax is **move 'delete' to "xpathList@action");** Example:

```
if ("parm/action = 'd'")
    move "0" to $index;
    for ($CCList in "CCList")
        move "1 + $index" to $index;
        if ("$CCList/id = parm/id")
            move 'delete' to "CCList[$index]@action";
            goto update;
        end-if;
    end-for;
end-if;
```

The following shows the resulting XML.

```
<root>
  <CCList>
    <id>9876538976</id>
    <balance>309.98</balance>
  </CCList>
  <CCList action="delete">
    <id>4321125899</id>
    <balance>87.45</balance>
  </CCList>
</root>
```

> **Note**
>
> Deleting a list instance through use of the action attribute is risky if iterative BO interactions are required. The XML document that contains the list instance to be deleted will not be altered after a successful BO interaction, which means the document will still contain the list instance even though it no longer exists. To solve this problem, it is essential to re-read the BO after any BO update where the action attribute of 'delete' has been used.

> **Note**
>
> An alternative to the delete attribute described here, is to use the BO action of replace. Manipulating a list to use the replace action avoids the problem described above concerning stale information in request documents post BO update.

### Functions for Processing a List

XPath provides several functions that are useful to process elements of a list including **count**, **sum** and **last**.


Copyright © 2026, Oracle and/or its affiliates.

The following examples are based on this sample XML document:

```
<xml>
  <ft>
    <type>bill</type>
    <date>20100101</date>
    <amt>30.30</amt>
    <cat>tax</cat>
  </ft>
  <ft>
    <type>adj</type>
    <date>20100301</date>
    <amt>20.20</amt>
    <cat>int</cat>
  </ft>
  <ft>
    <type>bill</type>
    <date>20100201</date>
    <amt>10.10</amt>
    <cat>tax</cat>
  </ft>
</xml>
```

The following is an example of a sum. The syntax is **move "sum(xpathList/element)" to $variable;** The example sums the total balance.

```
move "sum(ft/amt)" to $TotalBalance;
```

The following is an example of a sum using a filter to get a subtotal. The example sums the balance of the entries that have the ‘tax’ category.

```
move "sum(ft[cat='tax']/amt)" to $TaxBalance;
```

The following is an example of a count. The syntax is **move "count(xpathList)" to $variable;** The example finds the count of the number of FT entries in the list.

```
move "count(ft)" to $TranCount;
```

The following is an example of ‘last’, which is used to locate the last entry. The syntax is **move "last(xpathList)" to $variable;** The example finds the last amount in the FT list.

```
move "ft[last()]/amt" to $LastAmount;
```

### Declare and Invoke Schema Based Objects

You can invoke a business object, business service or service script within the edit data step. To support the dynamic invoke, a dynamic data area name can be declared.

The schema being declared may be a business object (BO) schema, a business service (BS) schema, a service script (SS) schema, data area (DA) schema or a UI map schema. The declare statement will differ based on the type of schema, but the syntax is analogous.

- **declareBO 'BO Name' | $variable | "xpath" as 'DynamicDataArea';**


Copyright © 2026, Oracle and/or its affiliates.

- **declareBS 'BS Name' | $variable | "xpath" as 'DynamicDataArea';**
- **declareSS 'SS Name' | $variable | "xpath" as 'DynamicDataArea';**
- **declareDA 'DA Name' | $variable | "xpath" as 'DynamicDataArea';**
- **declareMap 'Map Name' | $variable | "xpath" as 'DynamicDataArea';**

When invoking a BO, BS or SS, the name of the object can be specified as a literal or it can be a value contained within an element or a variable. For every Invoke, you must supply an XPath reference to a group name.

When invoking a business object, an action must be supplied. The syntax is **invokeBO 'BO Name' | $variable | "xpath" using "xpath" for action;**. The valid actions are as follows:

- **read**. This action reads the current view of the BO data.
- **add**. This action will add the object and read and return the resulting view of the BO.
- **fastAdd**. This action will add the object but does not perform a subsequent 'read' to return the resulting view of the BO. This option is better than add for performance purposes if there is no reason to re-read the record.
- **update**. This action will update the object and read and return the resulting view of the BO. This action executes a 'merge' of the information specified in the invoke statement's request XML document with existing BO data. Using this action allows the script to only indicate the elements that are changing.
- **fastUpdate**. This action will update the object but does not perform a subsequent 'read' to return the resulting view of the BO. This option is better than update for performance purposes if there is no reason to re-read the record.
- **delete**. This action deletes the object.
- **replace**. This action is an alternate to the update action. The replace action completely replaces existing BO data with the information in the request document. Typically, the replace action is used when a BO contains a list because it is easier to simply replace all instances of a list rather than attempt a list merge, which requires special logic to delete a list instance explicitly.

> **Note**
>
> The replace action must be used when using the UI map functionality to Upload a CSV File.

> **Note**
>
> The replace action is currently not supported for any maintenance object that is maintained in a 'fixed' page that uses a list metaphor to show all the records in the page at once. Currency is an example of this type of page.

Examples:

```
invokeBO 'BusinessObject' using "dataArea" for fastAdd;

invokeBO $variableBO using "dataArea" for fastUpdate;

invokeBO "daName/boElement" using "dataArea" for replace;
```


Copyright © 2026, Oracle and/or its affiliates.

The syntax of the invoke statements for both a business service and service script are similar. The BS / SS is specified along with the XPath reference to the group name:

- **invokeBS 'BS Name' | $variable | "xpath" using "xpath";**
- **invokeSS 'SS Name' | $variable | "xpath" using "xpath";**

The examples use the **invokeBS** statement but the statements are similar for the **invokeSS** statement.

```
invokeBS 'BusinessService' using "dataArea";

invokeBS $variableBS using "dataArea";

invokeBS "daName/bsElement" using "dataArea";
```

**BO Warnings.** Note that for BPA scripting, the **invoke** statements may also indicate how to handle warnings.

| Syntax | Description | Examples |
|---|---|---|
| `with warn asError` | Indicates that a warning should be treated as an error displayed in the UI map. The text **asError** is optional. | `invokeBO 'BusinessObject' using "dataArea" for add with warn asError;`<br><br>`invokeSS 'ServiceScript' using "dataArea" with warn;` |
| `with warn popup` | Indicates that a warning should be presented in the standard framework popup. In this scenario, the user is presented with standard OK and Cancel buttons. If the user clicks OK, it means that the process should continue. If the user clicks Cancel, the processing should discontinue. This is the recommended setting. | `invokeBS "daName/bsElement" using "dataArea" with warn popup;` |
| `with warn suppress` | Indicates that a warning should be suppressed. This is the default if no warning syntax is added to the **invoke** statement. | `invokeBS "daName/bsElement" using "dataArea" with warn suppress;`<br><br>`invokeSS 'ServiceScript' using "dataArea";` |

> **Note**
>
> For service scripts, all objects invoked from the service script will inherit their warning level. Therefore, if the service script is invoked **with warn**, all nested invoke statements will also be invoked **with warn**.


Copyright © 2026, Oracle and/or its affiliates.

For BPA scripts, there should also be logic following the invocation to handle errors and warnings (if **with warn as popup** is used). The system variable `$WARNING` is set to **true** if the user has clicked the Cancel button in the warning popup. If a map was shown, the logic should redisplay the map (where the warning message will be redisplayed.) This allows the user to potentially make changes and save again. If no map was shown prior to the warning, then the logic should terminate.

The system variable `$ERROR` indicates that an error was received. If a map was shown prior to the error, then the logic should redisplay the map where the error will be shown. If no map was shown, then the product provides a BPA Script `F1-HandleErr` that should be used to display the error. The following is an example of typical error handling logic.

```
invokeBO "F1-DetermineBo/output/bo" using "boSchema" for update with warn
popup;
if ("$WARNING")
   if ("map_schema/action = 'DEL'")
     terminate;
   else
     goto maintMap;
   end-if;
end-if;
if ("$ERROR")
  if ("map_schema/action != 'DEL'")
    goto maintMap;
  else
    transferControl 'F1-HandleErr';
  end-if;
end-if;
```

### System and Global Variables

The following tables highlight system and global variables available for script writing.

#### System Variables - All Script Types

The following system variables are available for all script types (service scripts, plug-in scripts, and BPA scripts).


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $BLANK | Reinfrest (sentsaingmpatym/sp.ouse/name) != $BLANK") |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | goto addSpouse;end-if; |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $CURRENT-DATE | Reprovesents the current date. For BPA scripts, this is the b |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | rowserdate. For server scripts this is the server date (an |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | disaffected by the system data overrides logic). |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $CURRENT-STD-DTTM | Represents the current data-time expressed in standard Data-Time |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | time (meaning without any adjustments for summertime/day |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | light savings time). |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $DEVICE-OS | Removes "$DEVICE-OS" device operating system. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $DEVICE-BROWSER | Represents the user's device browser. $DEVICE-BROWSER.$temp Device Bro |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | wser; |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description | Example |
| --- | --- | --- |
| $DEVICE-DISPLAY-TYPE | Represents the user's device screen display type whether i | move "$DEVICE-DISPLAY-TYPE" to $temp-Devi |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | Displays Type: Medium or Small size. Returned values m |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | ayb elikeoraDesktop,oraTabletandoraPhone. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $DEVICE-INFO | Provides the Combination of all three device properties (Description, Inventory, Fault) |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | DEVICE-OS, DEVICE-BROWSER and DEVICE-DISPLAY-TYPE) an |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | each property value is separated by semi-colon. |

### System Variables - BPA Scripts Only

The following system variables are only available / applicable for BPA script types.


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $DOUBLE_QUOTE | Represents a double-quote. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $SINGLE_QUOTE | Represents an apostrophe. "Use $SINGLE_QUOTE" to $tempField; |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $SPACE | Contains a single space value. Field; |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $SYSTEM-DATE | Reprovesents $SYSTEM-DATE. Note that this date is affected |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | dbythesystemdateoverridelogic) |

### System Variables - Server Scripts Only

The following system variables are only available / applicable for service script and plug-in script types.


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $ADDITIONAL-IP-INFO | Am HTTP "Pre" quest Dir Includes an "Pad Dir Fo" nal IP "ad" dre ms/s "he" quede |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | Example: Exam Profile/Table. This map is used for results, and by an implementation if there |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | is some information available on the proxy server or loadb |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | balancer, such as the originating IP address. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $CURRENT-DTTM | Represents the current date-time. $CURRENT-DTTM-time. $tempDateTime; |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $F1-INSTALLATION-TIMEZONE | Represents the time zone of the Oracle Fine Mortgage Instalation |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | Exampleption Zones; |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $LANGUAGE | Represents the language code that is used by Typical |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | ly this is the user's default language. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $PROCESS-DATE | Represents the process date. The process date differs from |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | m the current date because the process date will remain con |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | sistent throughout the duration of the process being exec |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | uted. For example, if a service script stores several busi |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | nessobjects–theprocessdateisinitializedatthestart |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | of the service script executed on each business object w |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | will have the same processed defaulted. The current date |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | , especially the current date time, will reflect the actual |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | time of processing. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $PROCESS-DTTM | Represents the Process Date-time. Note that the Process Date-time is not the Data Time. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | ate and time is initialized at the start of a particular pro |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | cessandwillnotreflectthe exactdateandtimeofanupdat |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | e. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $REQUESTING-IP-ADDRESS | Represents the IP address from the HTTP "request". Not the/that |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | Indicates/questions/requests through a proxy server or load balancer |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | ncer, this IP address is the IP address of the proxy or loa |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | dbalancer, not the IP address of the end user. Refer to the |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $ADDITIONAL-IP-INFO | variable for ip information. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| $USER | Represents the User ID of the user executing the script. For BPA script, the |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | a t n e e d t o k n o w t h e u s e r I D o f t h e c u r r e n t u s e r , r e f e r t o U s e f |


Copyright © 2026, Oracle and/or its affiliates.

| Variable | Description |
| --- | --- |
| | ultTipsandTricks. |

### Global Variables

BPA scripts and service scripts have access to the values defined in Global Context.

When a BPA script is launched from the user interface, these variables will be automatically initialized. They may be referenced with a single dollar sign in front of the field name. For example if PER_ID is a supported global variable, then $PER_ID can be referenced within the BPA script:

```
move "$PER_ID" to "schema/customerId";
```

For service scripts, global variables may only be referenced if the service script has been invoked directly from a BPA script or a zone on a portal. When a service script is invoked from a BPA script or portal zone, it will have access to the suite of global context variables populated in the UI session. For service scripting, the global fields must be prefixed by two dollar signs (instead of one like in BPA scripting). For example if PER_ID is a supported global context variable, then $$PER_ID can be referenced within the service script.

```
move $$PER_ID to "schema/customerId";
```

> **Note**
>
> As described in Context Variables, a service script may declare context variables that use the same two dollar sign syntax.


Copyright © 2026, Oracle and/or its affiliates.

### Perform Script and Transfer Control Statements

The edit data step supports functionality analogous to the Perform script step type and the Transfer Control step type. These are both applicable only to BPA scripts.

| Syntax | Valid Values | Comments |
|---|---|---|
| `performScript` | 'BPA Script Name' | Script to perform is explicitly provided. |
| | $Variable | Script to perform is found in a variable. |
| | "XPath" | Script to perform is found in an element, referenced by its XPath. |
| `transferControl` | Analogous to the **performScript** statement | |

> **Note**
>
> When the script named in the **performScript** statement has finished, control will be returned to the calling BPA script. When the script named in the **transferControl** statement has finished, you will not be returned to the calling script, complete control will be granted to the transferred to script.

### Navigate Statement

The edit data step supports functionality analogous to the Navigate to a page step type. This is applicable only to BPA scripts.

| Syntax | Valid Values | Comments |
|---|---|---|
| `navigate` | 'Navigation Code' | Navigation option is explicitly provided. |
| | $Variable | Navigation option is found in a variable. |
| | "XPath" | Navigation option is found in an element, referenced by its XPath. |

In addition, the edit data step supports the ability to indicate that the sidebar should be refreshed when navigating. This is only applicable to BPA scripts.

| Syntax | Valid Values |
|---|---|
| `navigateAndReloadDashboard` | Analogous to the **navigate** statement |

### Declare BO with BO Group

This statement is specific to BPA scripts that plan to use the base script Main BO Maintenance Processing (`F1_MainProc`) for its Generate Edit Map statements. This script expects that the data used to display in the map is within a **boGroup** tag.

| Syntax | Valid Values | Comments |
|---|---|---|
| `declareBOWithBOGroup` | 'BO Name' | BO is explicitly provided. |
| | $Variable | BO is found in a variable. |
| | "XPath" | BO is found in an element, referenced by its XPath. |

The following table highlights additional syntax for this statement.


Copyright © 2026, Oracle and/or its affiliates.

| Syntax | Valid Values |
|---|---|
| as | 'Dynamic Schema Name' |

Examples:

```
declareBOWithBOGroup 'BusinessObject' as 'newMapSchema';

declareBOWithBOGroup $variableBO as 'newMapSchema';

declareBOWithBOGroup "daName/boElement" as 'newMapSchema';
```

### Invoke Map Statement

The edit data step supports functionality analogous to the Invoke map step type. This is applicable only to BPA scripts.

| Syntax | Valid Values | Comments |
|---|---|---|
| invokeMap | 'Map Name' | UI Map is explicitly provided. |
| | $Variable | UI Map is found in a variable. |
| | "XPath" | UI Map is found in an element, referenced by its XPath. |

The following table highlights additional syntax for this statement.

| Syntax | Valid Values | Comments |
|---|---|---|
| using | "Data Area group name" | Indicates the data area to be passed to and from the server when rendering the HTML form associated with the map. |
| target | bpa | |
| | page | |
| | popup | |

Refer to the Invoke map step type for more information about the **target** values.

If the UI map is configured to return a value, then it can be evaluated using the $MAP-VALUE variable.

```
invokeMap 'UI Map' using "dataArea";

invokeMap $variableMap using "dataArea";

invokeMap "daName/mapElement" using "dataArea" target bpa;

// $MAP-VALUE is a variable returned by the invoked map.
if ("$MAP-VALUE='continue' ")
    goto 300;
else
    terminate;
end if;
```


Copyright © 2026, Oracle and/or its affiliates.

### Generate Edit Map Statements

The 'generate edit map' statements are used to dynamically generate and launch a UI edit map based on a schema definition. The schema used may be a BO schema, a BS schema, an SS schema or a DA schema. This is applicable only to BPA scripts. The generate statement will differ based on the type of schema, but the syntax is analogous.

| Syntax |
| --- |
| `generateBOEditMap` |
| `generateBSEditMap` |
| `generateSSEditMap` |
| `generateDAEditMap` |

The BO code / BS code / SS code / DA code may be specified using a literal (surrounded by single quotes), as a temporary variable or by referencing an XPath schema location (surrounded by double quotes).

The following table highlights additional syntax for this statement.

| Syntax | Valid Values | Comments |
| --- | --- | --- |
| `using` | "Data Area group name" | Indicates the data area to be passed to and from the server when rendering the HTML form associated with the map. |
| `target` | `bpa` | |
| | `page` | In Redwood view mode this option presents the form in the large drawer panel. |
| | `popup` | |
| | `sdrawer` | Presents the form in the small drawer panel. |
| | `mdrawer` | Presents the form in the medium drawer panel. |
| | `ldrawer` | Presents the form in the large drawer panel. |

The target values indicate where the generated map should be displayed as described in the Invoke map step type. If the UI map is configured to return a value, then it can be evaluated using the `$MAP-VALUE` variable.

The examples use the **generateBOEditMap** but the statements are similar for the other schema types.

```
generateBOEditMap 'BO Name' using "dataArea";

generateBOEditMap $variableMap using "dataArea";

generateBOEditMap "daName/mapElement" using "dataArea" target bpa;

generateBOEditMap "daName/mapElement" using "dataArea" target sdrawer;

generateBOEditMap "daName/mapElement" using "dataArea" target $drawerSize;
```


Copyright © 2026, Oracle and/or its affiliates.

```
// $MAP-VALUE is a variable returned by the invoked map.
if ("$MAP-VALUE='continue' ")
    goto 300;
else
    terminate;
end if;
```

### Terminate Statement

The edit data step supports functionality analogous to the Terminate step type.

The following is an example of a simple **terminate** step that will stop the script.

```
if ("not(parm/spouse/name)")
  terminate;
else
  goto addSpouse;
end-if;
```

The **terminate with error** statement is only available in a service script.

> **Note**
>
> For issuing an error from a BPA script refer to Useful Tips and Tricks.

| Syntax | Attributes | Comments |
|---|---|---|
| `terminate with error (x, y %n= element= )` | 'x' represents the message category | Required. |
| | 'y' represents the message number | Required. |
| | %n="Element XPath" or %n='literal' | Specify the substitution parameters supported by the message using either literal values or XPath references. When the value to substitute is an element within a list, use the XPath of the element within the list qualified by the list occurrence. |
| | element='Element XPath' or element=$variable | Optionally specify an element name within a UI map to highlight as part of the error. When the element in error is within a list, use the XPath of the element within the list qualified by the list occurrence. |

Example of a simple field:

```
if ("string(customer/lastName) = $BLANK")
  terminate with error (8000, 1001 %1="customer/lastName" %2='Last name
required' element='customer/lastName');
end-if;
```


Copyright © 2026, Oracle and/or its affiliates.

Example of terminate where the element to mark is in a list:

```
for ($list in "parm/hard/newBusinessObject/listName")
  if //** check some condition for elementName
      terminate with error (11000, 11000 %1="$list/elementName"
element='$list/elementName');
  end-if;
end-for;
```

> **NOT_SUPPORTED**
>
> For more information on presenting errors in a UI map, please refer to Display Errors.

### Invoking Groovy Code

If you have a plug-in script or a service script that mixes XPath scripting and Groovy scripting, the edit data step supports the ability to invoke Groovy code using the syntax **invokeGroovy 'method';** where 'method' is the name of a method defined in a **Groovy Members** step within the script. Only methods that do not receive arguments and return void can be invoked in this way. However, the method invoked from the edit data step can be supported by additional Groovy code in other **Groovy Members** step types.

Example Edit Data step:

```
invokeGroovy 'invoke';
```

Example Groovy Members step:

```
void invoke() {
  initParms()
  readBO()
  initConfig()
  retrieve()
  updateBO()
}
```

> **Note**
>
> As mentioned in Using Groovy Within Scripts, if this is a plug-in script and the code is only Groovy, you should use the **Groovy** script engine version rather than the above technique. This is considered more efficient as it avoids the conversion of the data to and from the XML interface.

### Debugging a BPA Script

If a BPA script has height greater than zero, then selected nodes of the script's data area can be displayed at runtime. The XML data is displayed during script execution within the BPA script's display area. Specify the XPath of an XML node from any of the BPA script's data areas, between the paired characters: '%+' and '+%'.

For example, the entire contents of the schema group node named 'input', and the specific contents of the schema element named 'output/status' will be displayed in the BPA script's


Copyright © 2026, Oracle and/or its affiliates.

display area. The debug text must be entered into the BPA script's text area and not within the script's edit data field. Debug text can be declared for any explicit step of the script.

```
display input: %+input+% , and output status: %+output/status+%
```

#### Useful Tips and Tricks

This section provides some suggestions about how to implement some common logic.

**Finding the Logged In User in a BPA**

For server based scripts, the $USER variable is populated with the current logged in user. This variable is not available for BPA scripts. The BPA can call the service script `F1-GetUser` if this information is needed.

**Issuing an Error from a BPA script**

For server based scripts, you use the **terminate with error** statement to return an error message to display to the user. The BPA can display that error by transferring control to `F1-HandleErr` as described in the Declare and Invoke Schema Based Objects. This is not supported in a BPA script. A technique that is available to a BPA for issuing an error is to call the business service `F1-RethrowError` passing the error details. This business service populates the appropriate system error fields needed to issue the error. Then transfer control to `F1-HandleErr`.

```
  if ("string($assignedToUser) != string(F1-GetUser/user)")
    declareBS 'F1-RethrowError' as 'errorMsg';
    move '11010' to "errorMsg/messageCategory";
    move '11511' to "errorMsg/messageNumber";
    move "$toDoEntryId" to "errorMsg/messageParameters/+parameters/
parameterValue";
    invokeBS 'F1-RethrowError' using "errorMsg";
    transferControl 'F1-HandleErr';
  end-if;
```

## Go To Steps

**Go to** steps allow you to jump to a step other than the next step. Additional fields required for **Go To** steps:

**Next Step** defines the step to which the script should jump.

## Height Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

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


Copyright © 2026, Oracle and/or its affiliates.

**Height** steps are used to change the height of the script area to be larger or smaller than the standard size.

The following additional fields are required for **Height** steps:

**Script Window Height** defines the number of **Pixels** or the **Percentage** (according to the **Height Unit**) that the script window height should be adjusted. The percentage indicates the percentage of the visible screen area that the script area uses. For example, a percentage value of `100` means that the script area will use the entire area.

> **Note**
>
> **Standard Number of Pixels**. The default number of pixels used by the script area is `75`.

> **Note**
>
> **Adjust script height in the first step**. If you want to adjust the height of the script area, it is recommendation to define the **height** step type as your first step. Otherwise, the script area will open using the standard height and then readjust, causing the screen to redisplay.

> **Note**
>
> **Hide script area**. You could use this type of step to set the height to `0` to hide the script area altogether. This is useful if the script does not require any prompting to the user. For example, perhaps you define a script to take a user to a page and with certain data pre-populated and that is all.

> **Note**
>
> **Automatically close script area**. If you want the script area to close when a script is completed, you could define the final step type with a height of 0.

## Input Data Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

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


Copyright © 2026, Oracle and/or its affiliates.

**Input data** steps cause the user to be prompted to populate an input field in the script area. The input value can be saved in a field on a page or in temporary storage. A **Continue** button always appears adjacent to the input field. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Input Data** steps:

**Destination Field Type** and **Destination Field Name** define where the input field will be saved. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Page Data Model**. Use this field type to put the input field into a field that resides on any of the tab pages in the object display area (i.e., the field populated doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed). Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Temporary Storage**. Use this field type to put the input field into temporary storage. Use **Field Name** to name the field in temporary storage. Refer to How To Name Temporary Storage Fields for more information.
- **User Interface Field**. Use this field type to put the input field into a field that resides on the currently displayed tab page. Note, if you want to execute underlying default logic, you must populate a **User Interface Field**. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

The **Prompt Values** grid may be used to define additional buttons. A separate button is displayed in the script area for each entry in this grid.

- **Prompt Text** is the verbiage to appear on the button. Refer to How To Substitute Variables In Text for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons.
- **Next Script Step** defines the step to execute if the user clicks the button.

## Invoke Business Object Steps

**Invoke business object** steps allow you to interact with a business object in order to obtain or maintain its information.

The following additional fields are required for **Invoke business object** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts. Refer to BO warnings for more information about handling warnings when invoking a BO.

**Group Name** references the data area to be passed to and from the server when communicating with the **Business Object**. Indicate the **Action** to be performed on the object when invoked. Valid values are **Add, Delete, Fast Add (No Read), Fast Update (No Read), Read, Replace, Update**.

> **NOT_SUPPORTED**
>
> Refer to BO Actions for more information about the various actions.


Copyright © 2026, Oracle and/or its affiliates.

The business object call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script, to determine the step to which control is passed given the outcome of the call.

**If Success, Go To** defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

**If Error, Go To** defines the step that is executed if the call returns on error. Please note that the error information is held in global variables. This field is only applicable to BPA scripts.

> **Note**
>
> **Error technique.** Let's assume a scenario where a business object is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

## Invoke Business Service Steps

**Invoke business service** steps allow you to interact with a business service.

The following additional fields are required for **Invoke business service** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts.

**Group Name** references the data area to be passed to and from the server when the **Business Service** is invoked.

The business service call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script, to determine the step to which control is passed given the outcome of the call.

**If Success, Go To** defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

**If Error, Go To** defines the step that is executed if the call returns on error. Please note that the error information is held in global variables. This field is only applicable to BPA scripts.

> **Note**
>
> **Error technique.** Let's assume a scenario where a business service is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Functions were implemented prior to the introduction of business services (BS), service scripts (SS) and business objects (BO). The functionality is still supported, but the recommendation for implementations going forward is to use a step that invokes one of the above configuration tool objects in a script rather than defining a function.

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

**Invoke function** steps may be used to retrieve or update data independent of the page currently being displayed. For example, if you design a script that takes different paths based on the customer's customer class, you could invoke a function to retrieve the customer's customer class.

> **NOT_SUPPORTED**
>
> You must set up a function before it can be referenced in a script. Refer to Maintaining Functions for the details.

The following additional fields are required for **Invoke Function** steps:

**Function** defines the name of the function. The function's **Long Description** is displayed below.

When a function is invoked, it will either be successful or return an error. The next two fields control the step to which control is passed given the outcome of the function call:

- **If Success, Go to** defines the step that is executed if the function is successful.
- **If Error, Go to** defines the step that is executed if the function returns on error. Refer to How To Use Constants In Scripts for a list of the global variables that are populated when a function returns an error.

> **Note**
>
> **Error technique.** If a function returns an error, we recommend that you invoke a step that transfers control to a script that displays the error message information and stops (note, the error information is held in global variables). You would invoke this script via a **Transfer Control**.

The **Send Fields** grid defines the fields whose values are sent to the function and whose field value source is not **Defined On The Function**. For example, if the function receives an account ID, you must define the name of the field in the script that holds the account ID.

- **Field** contains a brief description of the field sent to the function.


Copyright © 2026, Oracle and/or its affiliates.

- **Source Field Type** and **Mapped Field / Value** define the field sent to the function. Refer to the description of Source Field under How To Set Up Move Data Steps for a description of each field type.
- **Comments** contain information about the field (this is defined on the function).

The **Receive Fields** grid defines the fields that hold the values returned from the function. For example, if the function returns an account's customer class and credit rating, you must set up two fields in this grid.

- **Field** contains a brief description of the field returned from the function.
- **Destination Field Type** and **Mapped Field** define the field returned from the function. Refer to the description of Destination Field under How To Set Up Move Data Steps for a description of each field type.
- **Comments** contain information about how the field (this is defined on the function).

## Invoke Map Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

**Invoke map** steps are used to invoke a UI Map to display, capture and update data using an HTML form. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Invoke map** steps:

**Group Name** references the data area to be passed to and from the server when rendering the HTML form associated with the **Map**.

Use **Target Area** to designate where the map will be presented.

- Select **BPA Zone** if the map should be presented within the script area.

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

- Select **Page Area** if the map should be presented in the object display area, i.e. the frame typically used to house a maintenance page. In Redwood view mode this option presents the form in the large drawer panel.
- Select **Pop-up Window** if the map should be launched in a separate window.
- Select **Small, Medium, Large Drawer** if the map should be launched in a drawer panel.

The **Returned Values** grid contains a row for every button defined on the map.

- **Returned Value** is the value returned when the user clicks the button.


Copyright © 2026, Oracle and/or its affiliates.

- **Use as Default** can only be turned on for one entry in the grid. If this is turned on, this value's Next Script Step will be executed if the returned value does not match any other entry in the grid. For example, if the user closes a pop-up (rather than clicking a button), the default value will be used.
- **Next Script Step** defines the step to execute if the user clicks the button.

## Invoke Service Script Steps

**Invoke service script** steps allow you to execute a service script.

The following additional fields are required for **Invoke service script** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts.

**Group Name** references the data area to be passed to and from the server when the **Service Script** is invoked.

The service script call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script to determine the step to which control is passed given the outcome of the call.

**If Success, Go To** defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

**If Error, Go To** defines the step that is executed if the call returns on error. Please note that the error information is held in global variables. This field is only applicable to BPA scripts.

> **Note**
>
> **Error technique.** Let's assume a scenario where a service script is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

## Groovy Imports Steps

> **Note**
>
> **Server based scripts only.** This step type is not applicable to BPA scripts.

**Groovy Imports** step types provide a free format text area where you list the Groovy classes to be imported for use by the code in **Groovy Member** steps within the script.

Enter a description of the imports step in the **Text** field. Click the adjacent icon to open a window providing more space for entering text.

Enter the list of classes to import in the **Edit Data Text** field using the syntax **import 'class';** where 'class' is the fully qualified package name of the Groovy class. Click the adjacent icon to open a window providing more space for editing the text.


Copyright © 2026, Oracle and/or its affiliates.

When saving a script, if the system has been configured accordingly, when the system detects references to a Groovy class that is not explicitly listed in this step, the system warns the user and adds an explicit import for that class. (If no Groovy Import step type exists for the script, one will be added.) Users should review the imported class to ensure it is the expected class, in the case of multiple classes with a similar name.

> **Note**
>
> For security, the classes that may be imported are restricted to those allowed by the Framework. The warning and automatic update to the script is controlled by a property setting. Refer to the topic Using Groovy Within Scripts for more information.

## Groovy Library Interface Steps

> **Note**
>
> **Server based scripts only.** This step type is not applicable to BPA scripts.

**Groovy Library Interface** steps are only applicable to **Groovy Library Scripts**. They provide a free format text area where you can list the Groovy methods defined within the script that are available for use by other scripts.

Enter a description of the interface step in the **Text** field. Click the adjacent icon to open a window providing more space for entering text.

Enter the list of interface methods in the **Edit Data Text** field. Click the adjacent icon to open a window providing more space for editing the list.

> **Note**
>
> Every **Groovy Library Script** must have one and only one step of type **Groovy Library Interface**. The supporting code for the available methods is defined using one or more **Groovy Member** steps in the same script. Refer to the topic Using Groovy Within Scripts for more information.

## Groovy Member Steps

> **Note**
>
> **Server based scripts only.** This step type is not applicable to BPA scripts.

**Groovy Member** steps provide a free format text area where you can enter Groovy code.

Enter a description of the code block in the **Text** field. Click the adjacent icon to open a window providing more space for entering text.

Enter your code in the **Edit Data Text** field. Click the adjacent icon to open a window providing more space for editing the code.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> While it is possible to set up multiple steps of type **Groovy Member** the system treats these steps as a single class for compilation and execution purposes. Refer to the topic Using Groovy Within Scripts for more information.

**Useful APIs**

The following provides information about APIs that may be useful for Groovy programmers.

**Lookup object instantiation**

This API allows programmers to instantiate a lookup for subsequent use in Groovy code when there is no Java class generated for the lookup (as is often the case with custom lookups). Example:

```
Lookup customLookup = LookupHelper.getLookupInstance("ALG_ENTITY_FLG",
"CMAL");
```

## Label Steps

**Label** steps allow you to describe what the next step(s) are doing. Steps of this type are helpful to the script administrators when reviewing or modifying the steps in a script, especially when a script has many steps. When designing a script, the label steps enable you to provide a heading for common steps that belong together. The script tree displays steps of this type in a different color (green) so that they stand out from other steps.

There are no additional fields for **Label** steps.

## Mathematical Operation Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

**Mathematical operation** steps allow you to perform arithmetic on fields. You can also use this type of step to add and subtract days from dates. For example, you could calculate a date 7 days in the future and then use this value as the customer's next credit review date. The following additional fields are required for **Mathematical Operation** steps:

**Base Field Type** and **Base Field Name** define the field on which the mathematical operation will be performed. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Page Data Model**. Use this field type when the field resides on any of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Temporary Storage**. Use this field type when the field resides in temporary storage. You must initialize the temporary storage field with a Move Data step before performing mathematical operations on the field. Refer to How To Set Up Move Data Steps for more information.


Copyright © 2026, Oracle and/or its affiliates.

- **User Interface Field**. Use this field type when the field resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

**Math Operation** controls the math function to be applied to the **Base Field**. You can specify +, -, /, and *. Note, if the base field is a date, you can only use + or -.

**Math Field Type**, **Math Field Name** and **Math Field Value** define the field that contains the value to be added, subtracted, divided, or multiplied. The following points describe each field type:

- **Current To Do Information**. Use this field type when the value resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the value resides on any of the tab pages in the object display area. Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**.
- **Predefined Value**. Use this field type when the value is a constant. When this field type is used, use **Source Field Value** to define the constant value. Refer to How To Use Constants In Scripts for more information. Note, if you are performing arithmetic on a date, the field value must contain the number and type of **days**/ **months**/ **years**. For example, if you want to add 2 years to a date, the source field value would be **2 years**.
- **Temporary Storage**. Use this field type when the value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the value resides in a field on the current tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

## Move Data Steps

**Move data** steps allow you to move data (from a source to a destination). The following additional fields are required for **Move data** steps:

**Source Field Type**, **Source Field Name** and **Source Field Value** define what you're moving. The following points describe each field type:

- **Context Variable**. Use this field type in a plug-in or service script if the source value is a variable initiated in a higher level script. This is only applicable to Service Scripts and Plug-in Scripts.
- **Current To Do Information**. Use this field type when the source value resides on the current To Do entry. Refer to How To Use To Do Fields for instructions on how to define the appropriate **Field Name**. This is only applicable to BPA Scripts.
- **Data Area**. Use this field type when the field being compared is one that you put into one of the script's data areas in an earlier step. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Global Context**. Use this field type when the source value is a global context variable. This is only applicable to BPA Scripts.
- **Page Data Model**. Use this field type when the source value resides on any of the tab pages in the object display area (i.e., the source field doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed).


Copyright © 2026, Oracle and/or its affiliates.

Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.

- **Portal Context**. Use this field type when the source value is a variable in the portal context. This is only applicable to BPA Scripts.
- **Predefined Value**. Use this field type when the source value is a constant value defined in the script. When this field type is used, use **Source Field Value** to define the constant value. Refer to How To Use Constants In Scripts for instructions on how to use constants.

> **Note**
>
> **Concatenating fields together.** You can also use **Predefined Value** if you want to concatenate two fields together. For example, let's say you have a script that merges two persons into a single person. You might want this script to change the name of the person being merged out of existence to include the ID of the person remaining. In this example, you could enter a **Source Field Value** of `%ONAME` **merged into person %PERID** (where **ONAME** is a field in temporary storage that contains the name of the person being merged out of existence and **PERID** contains the ID of the person being kept). Refer to How To Substitute Variables In Text for a description of how you can substitute field values to compose the field value.

- **Temporary Storage**. Use this field type when the source value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field**. Use this field type when the source value resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.

**Destination Field Type** and **Destination Field Name** define where the source field will be moved. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Context Variable**. Use this field type in your plug-in or service script if you use a variable to communicate information to a lower level service script or schema. This is not applicable to BPA Scripts.
- **Data Area**. Use this field type when the destination field resides on one of the scripts data areas. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to How To Reference Fields In Data Areas for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model**. Use this field type when the destination field resides on any of the tab pages in the object display area (i.e., the field populated doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed). Refer to How To Find The Name Of Page Data Model Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.
- **Portal Context**. Use this field type when the destination to be updated is in the current portal context. This is only applicable to BPA Scripts.
- **Temporary Storage**. Use this field type when the destination field resides in temporary storage. Use **Field Name** to name the field in temporary storage. Use **Field Name** to name the field in temporary storage. Refer to How To Name Temporary Storage Fields for more information.


Copyright © 2026, Oracle and/or its affiliates.

- **User Interface Field**. Use this field type when the destination field resides on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**. This is only applicable to BPA Scripts.

> **Note**
>
> **Conditional field types.** The field types **Current To Do Information**, **Page Data Model** and **User Interface Field** are only applicable to BPA scripts.

## Navigate To A Page Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

**Navigate to a page** steps cause a new page (or tab within the existing page) to be displayed in the object display area. Steps of this type are a precursor to doing anything on the page. The following additional field is required for **Navigate to a page** steps:

**Navigation Option** defines the transaction, tab, access mode (add or change) and any context fields that are passed to the transaction in change mode. For example, if you want a script to navigate to Person - Characteristics for the current person being displayed in the sidebar, you must set up an appropriate navigation option. Refer to Defining Navigation Options for more information.

> **Note**
>
> **Navigating to a page in update mode.** Before you can navigate to a page in change mode, the page data model must contain the values to use for the navigation option's context fields. If necessary, you can move values into the page data model using a Move Data step first. For example, before you can navigate to a page in change mode with an account ID in context, you may need to move the desired account ID into the ACCT_ID field in the page data model. The actual field name(s) to use are listed as context fields on the navigation option.

## Perform Script Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

**Perform script** steps cause another BPA script to be performed. After the performed script completes, control is returned to the next step in the original script. You might want to think of the scripts referred to on steps of this type as "subroutines". This functionality allows you to encapsulate common logic in reusable BPA scripts that can be called from other BPA scripts. This simplifies maintenance over the long term.

The following additional field is required for **Perform script** steps:


Copyright © 2026, Oracle and/or its affiliates.

**Subscript** is the name of the script that is performed.

## Press A Button Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

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

**Press a button** steps cause a button or link text to be 'pressed' in the object display area, the application toolbar or the page title area. For example, you could use this type of step to add a new row to a person's characteristic (and then you could use a **Move Data** step to populate the newly added row with a given char type and value). The following additional fields are required for **Press a button** steps:

**Button Name** is the name of the button to be pressed. This button must reside on the currently displayed tab page (or in the application toolbar or page actions toolbar). Refer to How To Find The Name Of A Button for more information.

## Prompt User Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

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

**Prompt user** steps cause the user to be presented with a menu of options. The options can be presented using either buttons or in the contents of a drop down. You can also use steps of this type to pause a script while the user checks something out (and when the user is ready to continue with the script, they are instructed to click a prompt button). The following additional fields are required for **Prompt User** steps:

**Prompt Type** controls if the prompt shown in the script area is in the form of `Button(s)` or a **Dropdown**. Note, if you use a **Dropdown**, a Continue button appears adjacent to the dropdown in the script area when the step executes. The user clicks the Continue button when they are ready for the script to continue.

The **Prompt Values** grid contains a row for every value that can be selected by a user. Note, if you use a **Prompt Type** of `Button(s)`, a separate button is displayed in the script area for each entry in this grid.


Copyright © 2026, Oracle and/or its affiliates.

- **Prompt Text** is the verbiage to appear on the button or in the dropdown entry. Refer to How To Substitute Variables In Text for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons or dropdown entries.
- **Use As Default** can only be turned on for one entry in the grid. If this is turned on for a dropdown entry, this value is defaulted in the grid. If this is turned on for a button, this button becomes the default (and the user should just have to press `Enter` (or `space`) rather than click on it).
- **Next Script Step** defines the step to execute if the user clicks the button or selects the dropdown value.

## Set Focus To A Field Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

> **Caution**
>
> Guiding scripts that interact with the underlying page's fields and buttons are limited to surf over fixed pages only and are considered legacy functionality. This technique is not supported on top of portal pages and is planned to be deprecated in a future release.

**Set focus to a field** steps cause the cursor to be placed in a specific field on a page. A **Continue** button always appears in the script area when this type of step executes. The user may click the **Continue** button when they are ready for the script to continue. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Set focus to a field** steps:

**Destination Field Name** defines the field on which focus should be placed. This field must reside on the currently displayed tab page. Refer to How To Find The Name Of User Interface Fields for instructions on how to find the appropriate **Field Name**.

The **Prompt Values** grid may be used to define additional buttons. A separate button is displayed in the script area for each entry in this grid.

- **Prompt Text** is the verbiage to appear on the button. Refer to How To Substitute Variables In Text for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons.
- **Next Script Step** defines the step to execute if the user clicks the button.

## Terminate Steps

**Terminate** steps cause a server-based script to end processing successfully or issue an error.

The following additional fields are required for **Terminate** steps:


Copyright © 2026, Oracle and/or its affiliates.

**Error** indicates whether an error should be thrown or not. If error, **Error Data Text** must be specified, indicating the error message and any message substitution parameters. Refer to Edit Data Syntax the actual syntax of initiating an error message.

> **Note**
>
> The ability to terminate a step in error is only supported for server-based scripts.

### Transfer Control Steps

> **Note**
>
> **BPA scripts only.** This step type is only applicable to BPA scripts.

**Transfer control** steps cause the current BPA script to terminate and the control to pass to another BPA script. You might want to construct a BPA script with steps of this type when the script has several potential logic paths and you want to segregate each logic path into a separate BPA script (for ease of maintenance).

The following additional fields are required for **Transfer control** steps:

**Subscript** is the name of the script to which control is transferred.

## Additional Topics

The contents of this section provide additional information about steps.

- How To Find The Name Of User Interface Fields
- How To Find The Name Of Page Data Model Fields
- How To Find The Name Of A Button
- How To Substitute Variables In Text
- How To Use HTML Tags And Spans In Text Strings and Prompts
- How To Use Constants In Scripts
- How To Use Global Variables
- How To Name Temporary Storage Fields
- How To Work With Dates
- How To Use To Do Fields
- How To Reference Fields In Data Areas
- Script Engine Version 2 and Above Notes

### How To Find The Name Of User Interface Fields

This section is only relevant if you are using the legacy capability of BPA scripts to bring a user to a base fixed page and populate information into fields on that page.


Copyright © 2026, Oracle and/or its affiliates.

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

Follow these steps to find the name of a field that resides on a fixed page:

- Navigate to the page in question.
- Right click in the body of the page.
- Select **Inspect** from the pop-up menu to display the HTML.
- In the search look for the text "Widget Info". Note that if there are multiple sections on the page, you will find multiple "Widget Info" results. Based on the details shown, you are able to confirm which section you are looking for. For example, the following is an example from the Algorithm - Main page:

```
* Widget Info:
*     Widget_ID , Element Type - label info - label
*     ALG_CD, IT - CI_ALG$ALG_CD -  Algorithm Code
*     IM_ALG_CD, IM - $SEARCH_FOR_ALG_LBL -  Search for Algorithm
*     OWNER_FLG, IS - $OWNER_FLG -  Owner
*     DESCR50, IT - $DESCR50 -  Description
*     IM_Main_algtypCont, IM - $F1_SHOW_ALG_CTX -  Show Algorithm Context Menu
*     ALG_TYPE_CD, IT - $ALG_TYPE_CD -  Algorithm Type
*     IM_ALG_TYPE_CD, IM - $SEARCH_ALGTYPE_LBL -  Search for Algorithm Type
*     ALG_TYPE_DESCR50, PL - $DESCR50 -  Description
*     ALG_TYPE_LONG_DESC, IA - $DESCRLONG -  Detailed Description
*     IM_VerScroll_lftArrow, IM - $PREVIOUS_ALG_LBL -  Previous Version
*     ALG_VER$recordCount, SN - $OF_LBL -  Of
*     IM_VerScroll_rtArrow, IM - $NEXT_ALG_LBL -  Next Version
*     IM_ADD, IM - $ADD_ALG_LBL -  Add Version
*     IM_DEL, IM - $DELETE_ALG_LBL -  Delete  Version
*     ALG_VER$EFFDT, IT - $EFFDT -  Effective Date
*     L_OWNER_LBL, LP - $OWNER_FLG -  Owner
*     ALG_VER$OWNER_FLG, IS - $OWNER_FLG -  Owner
*     resultGrid, GD
```

The field names that you'll reference in your scripts are defined on the left side of the HTML (e.g., ALG_CD, DESCR50, ALG_TYPE_CD, etc.).

The names of fields that reside in scrolls are in a slightly different format. The following is an example of the HTML for the version scroll that appears on Algorithm - Main. Notice that the fields in the scroll are prefixed with the name of the scroll plus a $ sign. For example, the effective date is called ALG_VER$EFFDT.

```
*     IM_VerScroll_lftArrow, IM - $PREVIOUS_ALG_LBL -  Previous Version
*     ALG_VER$recordCount, SN - $OF_LBL -  Of
*     IM_VerScroll_rtArrow, IM - $NEXT_ALG_LBL -  Next Version
*     IM_ADD, IM - $ADD_ALG_LBL -  Add Version
*     IM_DEL, IM - $DELETE_ALG_LBL -  Delete  Version
*     ALG_VER$EFFDT, IT - $EFFDT -  Effective Date
*     L_OWNER_LBL, LP - $OWNER_FLG -  Owner
*     ALG_VER$OWNER_FLG, IS - $OWNER_FLG -  Owner
```

The names of fields that reside in grids are in a slightly different format. The following is an example of the HTML for the algorithm parameters grid that appears on Algorithm - Main. Notice that the fields in the grid are prefixed with the name of the grid plus a `:x$`. For example,


Copyright © 2026, Oracle and/or its affiliates.

the algorithm parameter label is called ALG_PARM:x$ALG_PARM_LBL. When you reference such a field in your script, you have the following choices:

- Substitute x with the row in the grid (and keep in mind, the first row in a grid is row 0 (zero); this means the second row is row 1).
- If you want to reference the "current row" (that is, the row in which the cursor will be placed), you can keep the x notation ( x means the "current row").

```
* Widget Info:
*     Widget_ID , Element Type - label info - label
*     ALG_PARM:x$ALG_PARM_LBL, IL - CI_ALG_TYPE_PRM_L$ALG_PARM_LBL -  Parameter
*     ALG_PARM:x$SEQNO, IL - $SEQNO -  Sequence
*     ALG_PARM:x$ALG_PARM_VAL, IA - $ALG_PARM_VAL -  Value
*
```

## How To Find The Name Of Page Data Model Fields

You find the name of a **Page Data Model** field in the same way described under How To Find The Name Of User Interface Fields. The only restriction is that you cannot refer to hidden / derived fields. However, you can refer to any of the object's fields regardless of the tab page on which they appear. For example, if you position the object display area to the Main tab of the Account transaction, you can reference fields that reside on all of the tab pages.

> ⚠ **Caution**
>
> If you populate a **Page Data Model** field, none of the underlying default logic takes place. For example, if you populate a customer contact's contact type, none of the characteristics associated with the customer contact type are defaulted onto the customer contact. If you want the underlying defaulting to take place, you must populate a **User Interface Field**.

## How To Find The Name Of A Button

This section is only relevant if you are using the legacy capability of BPA scripts to bring a user to a base fixed page and perform functionality by executing existing buttons on that page.

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

If you want a **Press a button** step to press a button or click a link in the application toolbar, use one of the following names:

| Button Name |
| --- |
| IM_GOBACK |
| IM_HISTORY |
| IM_GOFORWARD |
| IM_menuButton |


Copyright © 2026, Oracle and/or its affiliates.

| |
|---|
| IM_USER_HOME |
| IM_MY_PREF |
| IM_helpButton |

If you want a **Press a button** step to press a button in the page actions toolbar, use one of the following names:

| Button Name |
|---|
| IM_SAVE<br>IM_REFRESH<br>IM_CLEAR<br>IM_COPY |
| IM_DELETE |
| IM_ScrollBack |
| IM_ScrollForward |

The following buttons are also supported:

| Button Name | Comments |
|---|---|
| IM_TO_DO | |
| IM_CurrentTo Do | This navigates to the To Do Entry page for the user's current To Do. Refer to Working On A To Do for more information. |
| IM_MINIMIZE_DASHBOARD | Pressing this will collapse the sidebar. Note that when a script is finished, it will return the sidebar to the state it was when the script was launched. |
| IM_MAXIMIZE_DASHBOARD | Pressing this will expand the sidebar. Note that when a script is finished, it will return the sidebar to the state it was when the script was launched. |

Follow these steps to find the name of other buttons that reside in the object display area of a fixed page:

- Navigate to the page in question.
- Right click in the body of the page
- Select **Inspect** from the pop-up menu.
- In the search look for the text "Widget Info". Note that if there are multiple sections on the page, you will find multiple "Widget Info" results. Based on the details shown, you are able to confirm which section you are looking for.
- Iconized buttons (e.g., search buttons) are represented as HTML images and their field names are prefixed with **IM**. The following is an example of the HTML on the Algorithm - Main page (notice the **IM** fields for the iconized buttons).

```
 * Widget Info:
 *     Widget_ID , Element Type - label info - label
```


Copyright © 2026, Oracle and/or its affiliates.

```
    *     ALG_CD, IT - CI_ALG$ALG_CD -  Algorithm Code
    *     IM_ALG_CD, IM - $SEARCH_FOR_ALG_LBL -  Search for Algorithm
    *     OWNER_FLG, IS - $OWNER_FLG -  Owner
    *     DESCR50, IT - $DESCR50 -  Description
    *     IM_Main_algtypCont, IM - $F1_SHOW_ALG_CTX -  Show Algorithm Context
Menu
    *     ALG_TYPE_CD, IT - $ALG_TYPE_CD -  Algorithm Type
    *     IM_ALG_TYPE_CD, IM - $SEARCH_ALGTYPE_LBL -  Search for Algorithm
Type
```

- Transaction-specific actions buttons are represented as switches. The following is an example of the HTML on the Batch Job Submission - Main page (notice the **SW** fields for the buttons). Note, if you want to **Set focus** to such a field, you would move a **Predefined Value** of **TRUE** to the switch.

```
    *     ACTION_Q_SW, BU - $ACTION_Q_SW -  Duplicate & Queue
    *     ACTION_CANCEL_SW, BU - $CANCEL_SW -  Cancel
```

## How To Substitute Variables In Text

You can substitute field values into a step's text string. You do this by prefixing the field name whose value should be substituted in the string with a %. For example, the message, "On %COMPLETION_DTTM this bill was completed, it's ending balance was %ENDING_BALANCE" contains two substitution variables (the bill's completion date / time and the bill's ending balance).

To substitute the value of an element from a data area you need to reference its XPath location as follows: %=XPath=%. If you want to substitute the whole XML node, not just the value, you need to reference it as follows %+XPath+%.

Only fields linked to the current To Do and fields that reside in temporary storage and global variables can be substituted into a text string.

> **Note**
>
> You can substitute fields that reside in the User Interface or Page Data Model by first moving them into temporary storage (using a **Move data** step).

You can also substitute field values into the verbiage displayed in prompts using the same technique.

## How To Use HTML Tags And Spans In Text Strings and Prompts

You can use HTML tags in a step's text string. For example, the word "Continue" will be italicized in the following text string "PressContinue after you've selected the customer" (the  and  are the HTML tags used to indicate that the surrounded text should be italicized).

The following are other useful HTML tags:

- <br/> causes a line break in a text string. If you use <br/><br/> a blank line will appear.
- **** text  causes the surrounded text to be colored as specified (in this case, red). You can also use hex codes rather than the color name.


Copyright © 2026, Oracle and/or its affiliates.

Please refer to an HTML reference manual or website for more examples.

> **Note**
>
> Refer to Color Contrast for information about the use of the HTML color 'red' and its impact on accessibility.

## How To Use Constants In Scripts

Some steps can reference fields called **Predefined Values**. For example, if you want to compare an input value to the letter "Y", the letter Y would be defined as a Predefined Value's field value.

Special constants are used for fields defined as switches. When you move **TRUE** to a switch, it turns it on. When you move **FALSE** to a switch, it turns it off.

You can use a global variable as a Predefined Value. For example, if you wanted to move the current date to a field, you'd indicate you wanted to move a Predefined Value named %CURRENT_DATE.

## How To Use Global Variables

Some explicit steps can reference fields called **Predefined Values**. In addition to referencing an ad hoc constant value (e.g., the letter Y), you can also reference a global variable in such a field value. A global variable is used when you want to reference system data.

Note that when using the Edit Data step type, the variable available are slightly different. Refer to Edit Data Syntax for details.

The following global variables exist for BPA scripts:

| Variable Name | Comments |
|---|---|
| %PARM-<name> | This is the value of a parameter of that name passed in to the application when launched via the standard system URL. Refer to Launching A Script When Starting the System for more information on these parameters. |
| %PARM-NOT-SET | This is to be used to compare against **%PARM-**< > parameters to check if the parameter has been set or not when the application was launched. A parameter that has not been set would be considered equal to this global variable. It is recommended to compare parameters against this global variable before using them for the first time. |
| %BLANK | A constant that contains a blank value (no value). |
| %SPACE | A constant that contains a single space value. |
| %CURRENT-DATE | The current date as known by the browser, not the server. |
| %SYSTEM-DATE | The server date. Note that this date is affected by the system date override logic) |
| %SAVE-REQUIRED | A flag that contains an indication of whether the data on a page has been changed (and thus requires saving). You may want to interrogate this flag to force a user to save their work before executing subsequent steps. This flag will have a value of **TRUE** or **FALSE**. |


Copyright © 2026, Oracle and/or its affiliates.

| Variable Name | Comments |
|---|---|
| %NEWLINE | A constant that contains a new line character (carriage return). Upon substitution, a line break is inserted in the resultant text. |

> **Note**
>
> This constant does not have the desired effect when the resultant text is HTML. For example, a step's text and prompt strings. This is because HTML ignores special characters such as new lines. Refer to How To Use HTML Tags And Spans In Text to learn how to cause a line break in an HTML text.

To refer to a global context variable, use %FIELD_NAME. For example, if the field SP_ID is in the global context, you may reference %SP_ID to reference the ID of the service point currently in context. In addition, the following special values are supported:

| Variable Name | Comments |
|---|---|
| %CONTEXT-PERSONID | A constant that contains the ID of the current person. |
| %CONTEXT-ACCOUNTID | A constant that contains the ID of the current account. |
| %CONTEXT-PREMISEID | A constant that contains the ID of the current premise. |

In addition, if the script is invoking something else via one of the various “Invoke” step types and an error is returned, the following global variables contain information about the error:

| Variable Name | Comments |
|---|---|
| %ERRMSG-CATEGORY<br>%ERRMSG-NUMBER | The unique identifier of the error message number. |
| %ERRMSG-TEXT | The brief description of the error. |
| %ERRMSG-LONG | The complete description of the error. |

## How To Name Temporary Storage Fields

**Input Data** and **Move Data** steps can create fields in temporary storage. You specify the name of the temporary storage field in the step's **Field Name**. The name of the field must not begin with % and must not be named the same as the global variables. Besides this restriction, you can use any **Field Name** that's acceptable to JavaScript (i.e., you can name a field in temporary storage almost anything). Keep in mind that field names are case-sensitive.

## How To Work With Dates

Before we discuss how to work with dates in your scripts, we need to point out that there are two types of date fields: date-only and date-time. Date-only fields only contain a date. Date-time fields contain both a date and a time. The following topics describe how to work with dates on the various step types.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If you're working with a field that resides on the database (as opposed to a temporary storage field), the database field name will tell you what type of date it is: date-only fields are suffixed with **DT**, and date-time fields are suffixed with **DTTM**.

- Move Data
- Mathematical Operation
- Input Data

## Move Data

If you intend to use a **Move data** step to populate a *date-time* field, please be aware of the following:

- If the destination field resides in the *page data model*, the source field value must be in the format YYYY-MM-DD-HH.MM.SS or YYYY-MM-DD. If the field is in the format YYYY-MM-DD, the time of 12:00 am will be defaulted.
- If the destination field resides in the *user interface*, you must use two steps if you want to populate both date and time. To explain this, we'll assume the field you want to populate is called EXPIRE_DTTM:
  - First, you populate the date portion of the field. To do this, you'd move a date (this value can be in any valid date format that a user is allowed to enter) to a field called EXPIRE_DTTM_FWDDTM_P1. In other words, you suffix `_FWDDTM_P1` to the field name.
  - If you want to populate the time, you'd move the time (again, the field value can be in any format that a user could use to enter a time) to a field called EXPIRE_DTTM_FWDTTM_P2. In other words, you suffix `_FWDDTM_P2` to the field name.

If you intend to use a **Move data** step to populate a *date-only* field, please be aware of the following:

- If the destination field resides in the *page data model*, the source field value must be in the format YYYY-MM-DD.
- If the destination field resides in the *user interface*, the source field can be in any valid date format that a user is allowed to enter.

> **Note**
>
> **%CURRENT-DATE.** Keep in mind that the global variable%CURRENT-DATE contains the current date and you can move this to either a page data model, user interface, or temporary storage field. If you move %CURRENT-DATE to a temporary storage fields, it is held in the format YYYY-MM-DD.

## Mathematical Operation

If you intend to use a **Mathematical operation** step to calculate a date, you can reference both date-only and date-time fields. This is because mathematical operations are only performed against the date portion of date-time fields.


Copyright © 2026, Oracle and/or its affiliates.

Mathematical operations are limited to adding or subtracting days, months and years to / from a date.

> **Note**
>
> A useful technique to perform date arithmetic using the current date is to move the global variable%CURRENT-DATE to a temporary storage field and then perform the math on this field.

## Input Data

If you intend to use an **Input data** step on a *date-time* field, please be aware of the following:

- If the field resides in the *page data model*, the user must enter a value in the format YYYY-MM-DD-HH.MM.SS (and therefore we do not recommend doing this).
- If the field resides in the *user interface*, you must use two steps if you want to populate both date and time. To explain this, we'll assume the field you want to populate is called EXPIRE_DTTM:
  - First, you populate the date portion of the field. To do this, you'd input the date (this value can be in any valid date format that a user is allowed to enter) in a field called EXPIRE_DTTM_FWDDTM_P1. In other words, you suffix `_FWDDTM_P1` to the field name.
  - If you want to populate the time, you'd input the time (again, the field value can be in any format that a user could use to enter a time) in a field called EXPIRE_DTTM_FWDTTM_P2. In other words, you suffix `_FWDDTM_P2` to the field name.

If you intend to use an **Input data** step to populate a *date-only* field, please be aware of the following:

- If the field resides in the *page data model*, the user must enter a value in the format YYYY-MM-DD (and therefore we do not recommend doing this).
- If the field resides in the *user interface*, the user can enter any valid date format.

## How To Use To Do Fields

As described under Executing A Script When A To Do Entry Is Selected, you can set up the system to automatically launch a script when a user selects a To Do entry. These types of scripts invariably need to access data that resides on the selected To Do entry. The following points describe the type of information that resides on To Do entries:

- **Sort keys**. These values define the various ways a To Do list's entries may be sorted. For example, when you look at the bill segment error To Do List, you have the option of sorting the entries in error number order, account name order, or in customer class order. There is a sort key value for each of these options.
- **Message parameters**. These values are used when the system finds *%n* notation within the message text. The *%n* notation causes field values to be substituted into a message before it's displayed. For example, the message text **The %1 non-cash deposit for %2 expires on %3** will have the values of three fields merged into it before it is displayed to the user (%1 is the type of non-cash deposit, %2 is the name of the customer, and %3 is the expiration date of the non-cash deposit). Each of these three values is stored as a separate message parameter on the To Do entry.


Copyright © 2026, Oracle and/or its affiliates.

- **Drill keys**. These values are the keys passed to the page if a user drilled down on the entry (and the system wasn't set up to launch a script). For example, a To Do entry that has been set up to display an account on the account maintenance page has a drill key of the respective account ID.
- **To Do ID**. Every To Do entry has a unique identifier referred to as its To Do ID.

You can access this information in the following types of steps:

- **Move Data** steps can move any of the above to any data area. For example, you might want to move a To Do entry's drill key to the page data model so it can be used to navigate to a specific page.
- **Conditional Branch** steps can perform conditional logic based on any of the above. For example, you can perform conditional logic based on a To Do entry's message number (note, message numbers are frequently held in sort keys).
- **Mathematical Operation** steps can use the above in mathematical operations.

A To Do entry's sort key values are accessed by using a **Field Type** of **Current To Do Information** and a **Field Name** of `SORTKEY[index]`. Note, you can find an entry's potential sort keys by displaying the entry's To Do type and navigating to the Sort Keys tab. If you want to reference the first sort key, use an index value of `1`. If you want to use the second sort key, use an index value of 2 (and so on).

A To Do entry's drill key values are accessed by using a **Field Type** of **Current To Do Information** and a **Field Name** of `DRILLKEY[index]`. Note, you can find an entry's potential drill keys by displaying the entry's To Do type and navigating to the Drill Keys tab. If you want to use the first drill key, use an index value of `1`. If you want to use the second drill key, use an index value of 2 (and so on).

A To Do entry's message parameters are accessed by using a **Field Type** of **Current To Do Information** and a **Field Value** of `MSGPARM[index]`. Note, because a To Do type can have an unlimited number of messages and each message can have different parameters, finding an entry's message parameters requires some digging. The easiest way to determine these values is to display the To Do entry on To Do maintenance. On this page, you will find the entry's message category/number adjacent to the description. Once you know these values, display the message category/number on Message Maintenance. You'll find the message typically contains one or more %n notations (one for each message parameter). For example, the message text **The %1 non-cash deposit for %2 expires on %3** has three message parameters. You then need to deduce what each of the message parameters are. You do this by comparing the message on the To Do entry with the base message (it should be fairly intuitive as to what each message parameter is). If we continue using our example, `%1` is the non-cash deposit type, %2 is the account name, and %3 is the expiration date. You can access these in your scripts by using appropriate index value in `MSGPARM[index]`.

A To Do entry's unique ID is accessed by using a **Field Type** of **Current To Do Information** and a **Field Value** of `TD_ENTRY_ID`.

In addition, any of the above fields can be substituted into a text string or prompt. Simply prefix the To Do field name with a % as you would fields in temporary storage. For example, assume you want your script to display the following text in the script area: "ABC Supply does not have a bill cycle" (where ABC Supply is the account's name). If the first sort key linked to the To Do entry contains the account's name, you'd enter a text string of **%SORTKEY[1] does not have a bill cycle**.

## How To Reference Fields In Data Areas

Various step types involve referencing field elements residing in the script's data areas. To reference an element in a data area you need to provide its absolute XPath notation starting


Copyright © 2026, Oracle and/or its affiliates.

from the data area name. For example, use "F1-BatchControl/batchControl" to reference a top-level "batchControl" element in a script data area called "F1–BatchControl".

You don't have to type in long XPath notations. Use the **View Script Schema** hyperlink provided on the Script portal to launch the script's data areas schema.

**Figure Schema Viewer**

Doing this opens the schema viewer window where you can:

- Click on the field element you want to reference in your script step. The system automatically populates the text box on the top with the element's absolute XPath notation.
- Copy the element's XPath notation from the text box to your script.

You can also use the **View Data Area**, **View Service Script Data Area**, or **View Plug-In Script Data Area** links on the Data Areas tab page on the Script portal to the same effect. These open up the schema viewer for a specific data area respectively.

## Script Engine Version 2 and Above Notes

Scripting using the engine version 2 or above requires some extra syntax to take advantage of XPath 2 functionality. In general, any variable declared will be assumed to be a string. This means, that if you intend to construct a mathematical statement then it is necessary to explicitly declare the data type of variables as integers, numbers, or dates.

> **Note**
>
> Unless otherwise noted, all XPath examples in this topic are for the Version 1 engine – which means XPath 1. Statements that function using XPath 1 will not necessarily work for XPath 2. This is especially true when executing math, see below for examples.

### Date and Time Arithmetic

XPath date/time and interval data types support arithmetic operations ('+', '-', '*' etc.) and functions, which can be used for time calculations in the same way as '1 + xs:integer(value)' is used for numeric calculations.

Compare time duration:

```
if ("(xs:dateTime(fn:current-dateTime()) - xs:dateTime($updateDateTimeX))
   ge xs:dayTimeDuration(concat('PT', BO/hoursBetweenStatisticsUpdate, 'H'))")
   goto 60;
end-if;
```


Copyright © 2026, Oracle and/or its affiliates.

Compare one date to another:

```
if ("xs:date(parm/endDate) < xs:date(parm/startDate)")
   terminate with error (11108, 11507 element='endDate');
end-if;
```

Compare a date against today's date:

```
if ("xs:date(parm/startDate) <= xs:date($CURRENT-DATE)")
   terminate with error (11108, 11507 element='endDate');
end-if;
```

Calculate the end of month:

```
// covert to ISO
move "concat($year,'-',$mon2,'-01T00:00:00')" to $monthStart;

// calculate
move "xs:dateTime($monthStart) + xs:yearMonthDuration('P1M') -
xs:dayTimeDuration('P0DT1S')"
  to $monthEnd;

// convert from ISO to OUAF
move
"concat($year,'-',$mon2,'-',substring(string($monthEnd),9,2),'-23.59.59')"
to $endDateTime;
```

> **Note**
>
> XPath date/time/interval formats use the ISO standard, which needs to be converted to/from formats supported in the framework.

**Comparing Date/Times in String Format**

Any ISO-like string format for date/time preserves the YYYY MM DD HH MM SS sequence, which is zero-padded. Regardless of separators, this format will remain appropriate for comparison operations. In particular, date/time values in the framework format “YYYY-MM-DD.HH.MM.SS” can be used with “=”, “!=”, as well as “>”, “>=”, “<”, “<=” operators.

```
// retrieve framework date/time value
invokeBS 'CM-MAXMSRMT' using "CM-MAXMSRMT";
move "string(cm-MAXMSRMT/results[1]/measurementDateTime)" to $lastMsmtDT;

// construct another date/time
move "concat($year,'-01-01-00.00.00')" to $startDateTime;

// compare using string operators
if ("$lastMsmtDT >= $startDateTime")
    move "substring($lastMsmtDT,1,4)" to $latestMsrmtYear;
```


Copyright © 2026, Oracle and/or its affiliates.

### Converting Date/Times Between Framework and ISO

Conversion of date/time from framework format to ISO is only necessary for date/time arithmetic. Comparisons can be done with the framework format directly. The only difference between the framework format and ISO date/time formats is in the separators:

Framework: “YYYY-MM-DD.HH.MM.SS”

ISO: “YYYY-MM-DDTHH:MM:SS”

Example of converting from the framework format to ISO:

```
move "concat(substring($ouafDT, 1, 10), 'T', translate(substring($ouafDT,
12),'.',':'))" to $isoDT;
```

Example of converting from ISO to the framework format:

```
move "concat(substring($isoDT, 1, 10), '.', translate(substring($isoDT,
12),':','.'))" to $ouafDT;
```

### Round Money With a Dynamic Currency Scale

Because different currencies support a different number of decimals, the framework provides an API for rounding a monetary amount based on a given currency.

```
move "parm/amount" to $qnty;
move "currency/decimals" to $decimals;
move "fn:round(xs:decimal($qnty) * math:exp10(xs:double($decimals)))
  div math:exp10(xs:double($decimals))" to "parm/roundedAmount";
```

### Looping through Sequences

In XPath 2 it is possible to organize a for-loop over a sequence of integers, not only a node list.

This example shows a loop over a range of months. This is a sequence-forming construct in XPath. The XPath node list, which we are familiar with, is just another type of sequence.

```
for ($month in "1 to 12")
```

This example shows a loop over a give range of years in descending order:

```
for ($year in "fn:reverse(parm/startYear to parm/endYear)")
     move "concat($year,'-01-01-00.00.00')" to $startDateTime;
     move "concat($year,'-12-31-23.59.59')" to $endDateTime;
     ...
```

This example shows a loop through a node list using ‘index’, so that other node lists can be accessed:

```
for ($idx in "1 to count(parm/touData/touList)")
     move "parm/touData/touList[$idx]" to $tou;   // access any list with
this index
```


Copyright © 2026, Oracle and/or its affiliates.

The above syntax can be used as an elegant alternative to maintaining indices separately, for example instead of the following:

```
move “0” to $idx;
for ($item in "parm/touData/touList")
     move “1 + xs:integer($idx)” to $idx;
```

### String Padding and Decimal Formatting

This is used with specific input formats or output formatting. It is applicable to zero, space and other types of padding.

This example shows prefixing for date/time components, for example producing “2010-01-02” instead of “2010-1-2”.

```
move "substring(concat('0',string($month)), string-length(string($month)),
2)" to $mon2;
```

This example shows suffixing for adding decimal zero-padded alignment, for example producing “12.30” and “4.00” instead of “12.3” and “4”. The example performs 3 tasks: rounding to 2 decimals, inserting a period if necessary, and zero padding.

```
// round and zero-pad to 2 decimals
move "$item/amount" to $qty;
move "fn:round(xs:double($qty) * 100) div 100" to $qty;
move "string($qty)" to $qty;
move "concat(substring-
before(concat($qty,'.'),'.'),'.',substring(concat(substring-
after($qty,'.'),'00'),1,2))" to $qty;
```

### Ternary Operation

This makes a choice between values based on a condition, so that it could be used in a single expression instead of an if/else block. It is know in C/C++ as ‘cond ? value1 : value2’ or in BASIC as ‘IFF(cond, value1, value2)’. In XPath the syntax is: "if (cond) then value1 else value2". Note this is not the top-level scripting if-statement block.

In XPath this is an expression, which can be combined with other expressions. In scripting it can be used as:

```
move "if (string(D1-UnitOfMeasure/measuresPeakQuantity) = 'D1MP') then 'D1MX'
else 'D1SM' " to $func;
```

### Pipeline Processing

In scripting, it is not easy to create a simple reusable piece of code as there are no local functions, and a separate script call is a coding overhead and requires packing/unpacking parameters. To avoid copying and pasting the same code block between similar script stages, consider ‘pipelining’, which is breaking the overall process into separate top-level steps, some of which could be shared between alternating paths. This is common for parameter preparation and output formatting. An intermediate result between stages can be stored in a “parm” substructure.


Copyright © 2026, Oracle and/or its affiliates.

| Instead of this code: | Consider this alternative: |
| --- | --- |
| `if ("type = A")`<br>`  prepare params ...`<br>`  call services for A ...`<br>`  format output ...`<br>`end-if;`<br>`if ("type = B")`<br>`  prepare params ...`<br>`  call services for B ...`<br>`  format output ...`<br>`end-if;` | `prepare params ...`<br><br>`if ("type = A")`<br>`  call services for A ...`<br>`end-if;`<br>`if ("type = B")`<br>`  call services for B ...`<br>`end-if;`<br><br>`format output ...` |

**XPath 2 Functions**

Script engine versions 2 and above support XQuery 1.0 Functions and Operators, and the XQuery 1.0 standard itself with some minor limitations. Below are the URLs to both specifications. The first link has the functions/operators available to use from XQuery.

- http://www.w3.org/TR/xpath-functions/
- http://www.w3.org/TR/xquery/

The following can only access local file systems. (For other protocols like http they will return an empty sequence):

- fn:doc
- fn:collection

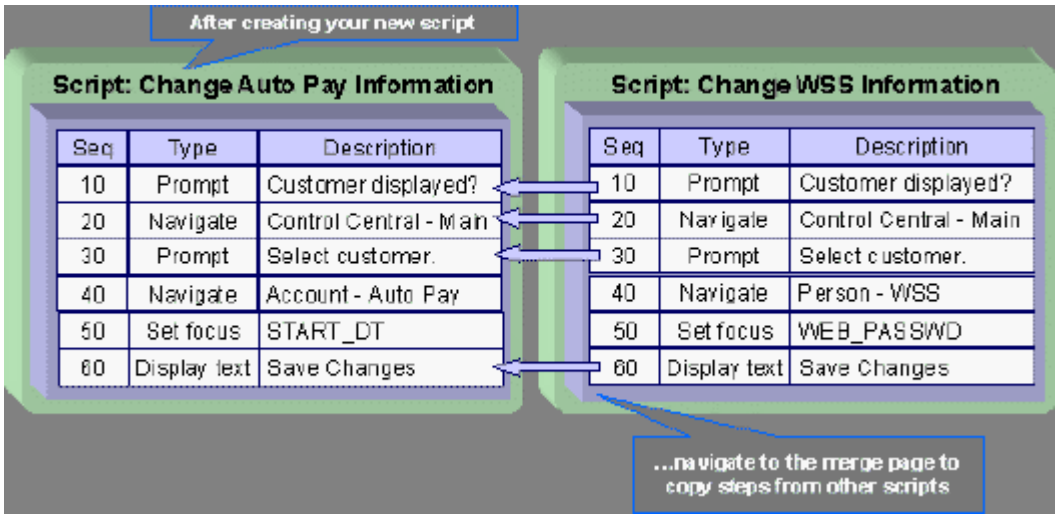
# Merging Scripts

Use the Script Merge page to modify an existing script by copying steps from other scripts. The following points summarize the many diverse functions available on the Script Merge transaction:

- You can use this transaction to renumber steps (assign them new sequence numbers).
- You can use this transaction to move a step to a different position within a script. When a step is moved, all references to the step are changed to reflect the new sequence number.
- You can use this transaction to delete a step.
- You can use this transaction to copy steps from other scripts. For example:
  - You may want to create a script that is similar to an existing script. Rather than copying all the information from the existing script and then removing the inapplicable steps, this page may be used to selectively copy steps from the existing script to the new script.
  - You may have scripts that are very similar, but still unique. You can use this transaction to build large scripts from smaller scripts. In this scenario, you may choose to create special 'mini' scripts, one for each of the various options that may make a script unique. Then, you could use the script merge page to select and merge the mini scripts that are applicable for a main script.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **The target script must exist prior to using this page.** If you are creating a new script, you must first create the Script and then navigate to the merge page to copy step information.




> **Note**
>
> **Duplicate versus Merge**. The Script page itself has duplication capability. You would duplicate a script if you want to a) create a new script and b) populate it with *all* the steps from an existing script.

- Script Merge

## Script Merge

Open **Admin**, and then **System**, and then **Script Merge** to open this page.

**Description of Page**

For **Original Script**, select the target script for merging steps.

For **Merge From Script**, select the template script from which to copy the steps.

> **Note**
>
> You may only copy steps from one Merge From script at a time. If you want to copy steps from more than one script, select the first Merge From script, copy the desired steps, save the original script, and then select the next Merge From script.

The left portion of the page displays any existing steps for the **Original Script**. The right portion of the page displays the existing steps for the **Merge From Script**.

You can use the **Copy All** button to copy all the steps from the **Merge From** script to the **Original** script. If you use **Copy All**, the steps are added to the end of the original script.


Copyright © 2026, Oracle and/or its affiliates.

Each time you save the changes, the system renumbers the steps in the original script using the **Start From Sequence Number** and **Increment By**.

**Merge Type** indicates **Original** for steps that have already been saved in the original script or **Merge** for steps that have been merged, but not yet saved. The **Sequence**, **Step Type** and **Description** for each step are displayed.

The topics that follow describe how to perform common maintenance tasks:

### Resequencing Steps

If you need to resequence the steps:

- Use the up and down arrows in the Original Script grid to reorder the steps.
- Make any desired changes to the **Start From Sequence Number** or **Increment By**.
- Click Save.

The steps are given new sequence numbers according to their order in the grid.

### Removing a Step from Script

If you want to remove a record linked to the Original script, click the delete button to the left of the record.

For example, to remove the **Reset existing bundle XML** step, click the icon.

| | | Merge Type | Sequence | Step Type | Description |
|---|---|---|---|---|---|
| | | Original | 10 | Edit data | Edit data - Check that the BO is an Export Bundle |
| | | Original | 20 | Edit data | Edit data - Read the Bundle |
| | | Original | 30 | Edit data | Edit data - Reset existing bundle XML |
| | | Original | 40 | Edit data | Edit data - Create new bundle XML |

After removal, the grid displays:

| | | Merge Type | Sequence | Step Type | Description |
|---|---|---|---|---|---|
| | | Original | 10 | Edit data | Edit data - Check that the BO is an Export Bundle |
| | | Original | 20 | Edit data | Edit data - Read the Bundle |
| | | Original | 40 | Edit data | Edit data - Create new bundle XML |

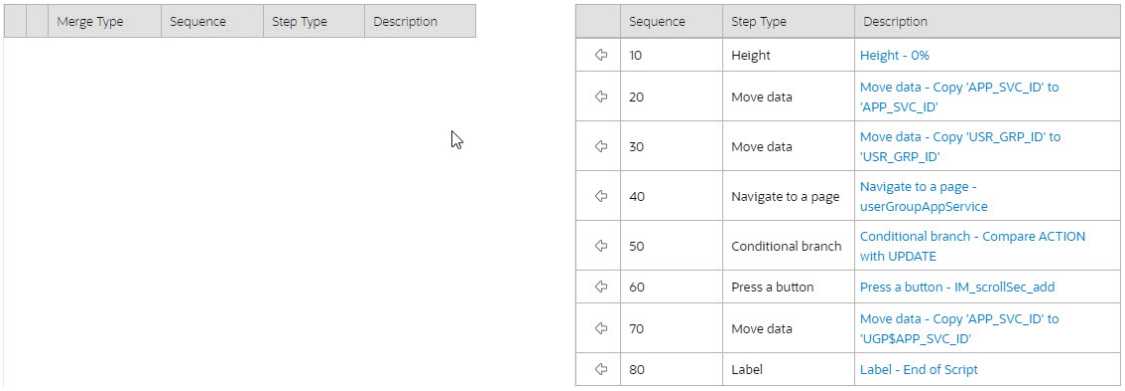
> **Note**
>
> You cannot delete a step that is referenced by other steps unless you also delete the referencing steps, such as **Go to step** or **Prompt** type steps. The system informs you of any missing referenced steps when you attempt to save the original script.


Copyright © 2026, Oracle and/or its affiliates.

### Adding a Step to a Script

You can move any of the steps from the Merge From script to the Original Script by clicking the left arrow adjacent to the desired step. Once a record is moved it disappears from the Merge From information and appears in the Original information with the word **Merge** in the Merge Type column.

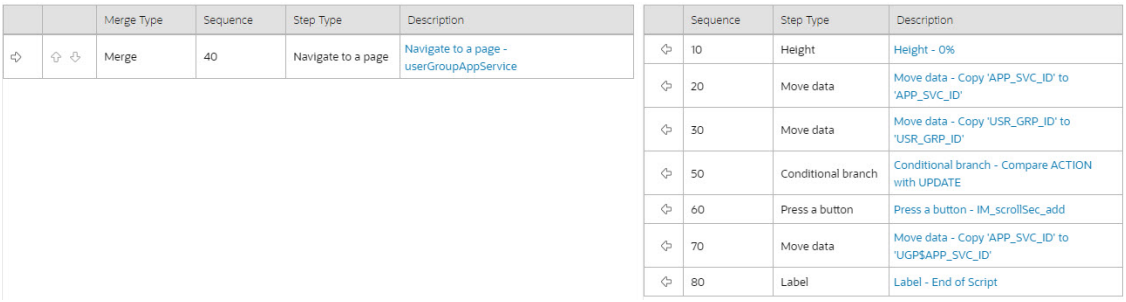
For example, to copy the **Navigate to a page** step, click the left arrow.



| | Merge Type | Sequence | Step Type | Description |
|---|---|---|---|---|

| | Sequence | Step Type | Description |
|---|---|---|---|
| ⇦ | 10 | Height | Height - 0% |
| ⇦ | 20 | Move data | Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID' |
| ⇦ | 30 | Move data | Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID' |
| ⇦ | 40 | Navigate to a page | Navigate to a page - userGroupAppService |
| ⇦ | 50 | Conditional branch | Conditional branch - Compare ACTION with UPDATE |
| ⇦ | 60 | Press a button | Press a button - IM_scrollSec_add |
| ⇦ | 70 | Move data | Move data - Copy 'APP_SVC_ID' to 'UGP$APP_SVC_ID' |
| ⇦ | 80 | Label | Label - End of Script |

The step is moved to the left portion of the page.

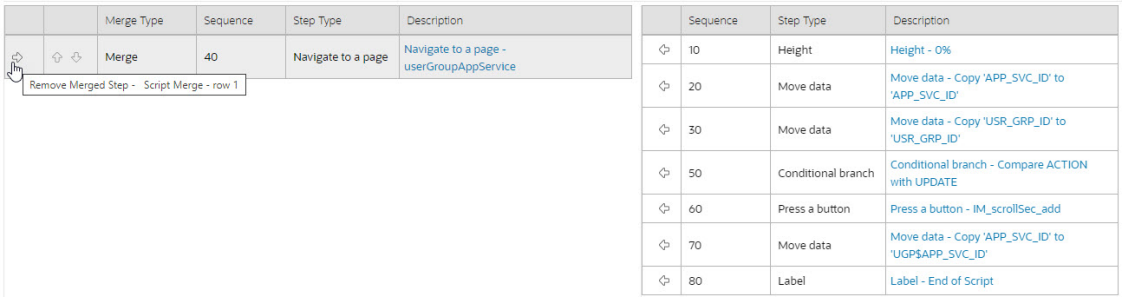


| | | Merge Type | Sequence | Step Type | Description |
|---|---|---|---|---|---|
| ⇨ | ⇧ ⇩ | Merge | 40 | Navigate to a page | Navigate to a page - userGroupAppService |

| | Sequence | Step Type | Description |
|---|---|---|---|
| ⇦ | 10 | Height | Height - 0% |
| ⇦ | 20 | Move data | Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID' |
| ⇦ | 30 | Move data | Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID' |
| ⇦ | 50 | Conditional branch | Conditional branch - Compare ACTION with UPDATE |
| ⇦ | 60 | Press a button | Press a button - IM_scrollSec_add |
| ⇦ | 70 | Move data | Move data - Copy 'APP_SVC_ID' to 'UGP$APP_SVC_ID' |
| ⇦ | 80 | Label | Label - End of Script |

> **Note**
>
> If you add a step, such as **Go to step** or **Prompt** type steps, that references other steps, you must also add the referenced steps. The step references are updated to use the new sequence numbers when you save the original script. The system informs you of any referenced steps that haven't been added when you attempt to save the original script.

### Removing an Uncommitted Step from a Script



| | | Merge Type | Sequence | Step Type | Description |
|---|---|---|---|---|---|
| ⇨ | ⇧ ⇩ | Merge | 40 | Navigate to a page | Navigate to a page - userGroupAppService |

Remove Merged Step - Script Merge - row 1

| | Sequence | Step Type | Description |
|---|---|---|---|
| ⇦ | 10 | Height | Height - 0% |
| ⇦ | 20 | Move data | Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID' |
| ⇦ | 30 | Move data | Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID' |
| ⇦ | 50 | Conditional branch | Conditional branch - Compare ACTION with UPDATE |
| ⇦ | 60 | Press a button | Press a button - IM_scrollSec_add |
| ⇦ | 70 | Move data | Move data - Copy 'APP_SVC_ID' to 'UGP$APP_SVC_ID' |
| ⇦ | 80 | Label | Label - End of Script |


Copyright © 2026, Oracle and/or its affiliates.

# Maintaining Functions

> **Note**
>
> Functions were implemented prior to the introduction of business services (BS), service scripts (SS) and business objects (BO). The functionality is still supported, but the recommendation for implementations going forward is to use one of the above configuration tool objects in a script rather than defining a function. The documentation has not been updated throughout this section to highlight where BS, SS or BO could be used to perform the equivalent logic.

**Invoke function** steps may be used to retrieve or update data independent of the page currently being displayed. For example, if you design a script that takes different paths based on the customer's customer class, you could invoke a function to retrieve the customer's customer class. Doing this is much more efficient than the alternative of transferring to the account page and retrieving the customer class from the Main page.

An **Invoke function** step retrieves or updates the relevant data by executing a service (on the server). These types of steps do not refer to the service directly. Rather, they reference a "function" and the function, in turn, references the service.

> **Note**
>
> **Functions are abstractions of services.** A function is nothing more than meta-data defining the name of a service and how to send data to it and retrieve data from it. Functions allow you to define a scriptwriter's interface to services. They also allow you to simplify a scriptwriter's set up burden as functions can handle the movement of data into and out of the service's XML document.

The topics in this section describe how to set up a function.

> **Note**
>
> **You can retrieve data from all base-package objects.** If you know the name of the base-package "page" service used to inquire upon an object, you can retrieve the value of any of its fields for use in your scripts. To do this, set up a function that sends the unique identifier of the object to the service and retrieves the desired fields from it.

- Function - Main
- Function - Send Fields
- Function - Receive Fields

## Function - Main

Use this page to define basic information about a function. Open this page using **Admin**, and then **System**, and then **Function.**

**Description of Page**


Copyright © 2026, Oracle and/or its affiliates.

Enter a unique **Function** code and **Description** for the function.

Use the **Long Description** to describe, in detail, what the function does.

Define the **Internal Service** that the function invokes.

> **Note**
>
> Only page services can be invoked.

The tree summarizes the following:

- The fields sent to the service. You can use the hyperlink to transfer to the **Send Fields** tab with the corresponding field displayed.
- The fields received from the service. You can use the hyperlink to transfer to the **Receive Fields** tab with the corresponding field displayed.
- Scripts that reference the function. You can use the hyperlink to transfer to the script page.

## Function - Send Fields

Use this page to add or update the fields sent to the service. Open this page using **Admin**, and then **System**, and then **Function** and then navigate to the **Send Fields** tab.

> **Note**
>
> **Displaying a specific field.** Rather than scrolling through each field, you can navigate to a field by clicking on the respective node in the tree on the Main tab. Also note, you can use the Alt+right arrow and Alt+left arrow accelerator keys to quickly display the next and previous entry in the scroll.

> **Note**
>
> **You're defining the service's input fields.** On this tab, you define which fields are populated in the XML document that is sent to the service. Essentially, these are the service's input fields.

### Description of Page

Use **Sequence** to define the order of the **Send Fields**.

Enter a unique **Function Field Name** and **Description** for each field sent to the application service. Feel free to enter **Comments** to describe how the field is used by the service.

Use **Field Value Source** to define the source of the field value in the XML document sent to the service:

- If the field's value is the same every time the function is invoked, select **Defined On The Function**. Fields of this type typically are used to support "hard-coded" input values (so that the scriptwriter doesn't have to populate the field every time they invoke the function). Enter the "hard-coded" **Field Value** in the adjacent field.
- If the field's value is supplied by the script, select **Supplied By The Invoker**. For example, if the function retrieves an account's customer class, the script would need to supply the


Copyright © 2026, Oracle and/or its affiliates.

value of the account ID (because a different account ID is passed each time the function is invoked). Turn on **Required** if the invoker must supply the field's value (it's possible to have optional input fields).

Regardless of the Field Value Source, use **XML Population Logic** to define the XPath expression used to populate the field's value in the XML document sent to the service.

## Function - Receive Fields

Use this page to add or update the fields received from the service. Open this page using **Admin**, and then **System**, and then **Function** and then navigate to the **Receive Fields** tab.

> **Note**
>
> **Displaying a specific field.** Rather than scrolling through each field, you can navigate to a field by clicking on the respective node in the tree on the Main tab. Also note, you can use the Alt+right arrow and Alt+left arrow accelerator keys to quickly display the next and previous entry in the scroll.

> **Note**
>
> **You're defining the application service's output fields.** On this tab, you define which fields are populated in the XML document that is received from the service. Essentially, these are the service's output fields.

**Description of Page**

Use **Sequence** to define the order of the **Receive Fields**.

Enter a unique **Function Field Name** and **Description** for each field received from the service. Feel free to enter **Comments** to describe the potential values returned from the service.

Turn on **Required** if the invoker must use the field.

Regardless of the Field Value Source, use **XML Population Logic** to define the XPath expression used to retrieve the field's value from the XML document received from the service.

> **Note**
>
> **Fields in multiple lists.** If a field within the list is unique across the service, the generic list reference is sufficient for the XML population logic. However, if the field you are trying to reference is in multiple lists, the XPath must include the list name. For example, instead of `/pageBody/list/listBody/field[@name='FIELD_NAME']`, the XPath Population Logic must read `/pageBody/list[@name='LIST_NAME']/listBody/field[@name='FIELD_NAME']`.

# Maintaining Managed Content

The Managed Content object is used to store various types of text content. Some content is used internally such as Allowlists. Other content may be used by implementations such as XSL


Copyright © 2026, Oracle and/or its affiliates.

files used for communicating messages with an external system, JavaScript include files, and CSS files. These files may then be maintained in the same manner as the HTML in UI Maps.

Open this page using **Admin**, and then **System**, and then **Managed Content**.

Use the **Managed Content Query** portal to search for an existing managed content entry. Once the managed content is selected, you are brought to the maintenance page to view and maintain the selected record.

Most fields on the page are self explanatory.

Use **Managed Content Type** to indicate the type of content.

- **Allowlist** is used to capture a list of supported values for given functionality. For example, the product provides allowlists for SQL functions and valid HTML tags.
- **XSLTC** is used to reference XSL files. These may be used internally for user interface referencing. Additionally, the XSLs may be referenced for sending messages to external system using outbound message and for receiving messages in SOAP as inbound messages.

> **Note**
>
> The value F1X (XSLT) is no longer supported and will be deprecated in a future release. Other managed content values are not currently supported by any current framework functionality and are reserved for future use.

Use the **Schema** tab to view and maintain the contents of the managed context text.

# Defining Entity Tags

This portal is used to maintain an entity tag along with the list of entities associated with it. Entity tagging supports grouping of configuration, demo and test related entities so they can better describe a logical function. Each group is represented by a unique tag name and entities may be associated with multiple tags.

An entity tag may explicitly restrict the type of entities it can be associated with by referencing a specific maintenance object as part of the entity tag record.

You can access the portal from the**Admin Menu**, and then **Implementation Tools**, and then **Entity Tag**. You are brought to a query portal with options for searching for a specific tag name. Once an entity tag has been selected you are brought to the maintenance portal to view and maintain the selected record.

A Tags sidebar zone allows you to review, add and remove tags for the current entity you review. For configuration type of entities, you may use this portal to tag multiple entities at once.

The following zones may appear as part of the portal's **Main** tab page

- **Entity Tag.** This zone provides information about the selected entity tag.
- **Tagged Entities.** This zone lists entities associated with the selected entity tag. You may also use the zone to remove entities from the list.
- **Tag Entities.** Use this zone to add configuration entities to the selected tag.


Copyright © 2026, Oracle and/or its affiliates.

# Advanced Schema Topics

The topics in this section describe some advanced information related to schemas used for business objects, business services, service scripts and UI maps.

- Schema Nodes and Attributes
- Schema Designer
- Schema Viewer

## Schema Nodes and Attributes

For business object definition, the purpose of the schema is to create a link between the schema and a maintenance object. For business service definition you are specifying the link between the schema and a service (either a general service, search service, or a maintenance object service). For service script definition, you are defining the API for passing information to and from the script. The following documentation is a complete list of the XML nodes and attributes available to you when you construct a schema.

> **Note**
>
> The definition of a schema may include user interface related attributes that represent a separate aspect of the schema for presentation purposes. These attribtutes belong to a separate name space known as UI Hints and as such are not part of the schema's data structure. Refer to Understanding UI Hints for more information.

### The Four Element Types

A schema element can be one of four different structure types. Note that there are two classes of element types: the structural nodes group and list, and the data containing nodes of field and raw.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| `type=` | `"field"` | The field type is the default type for any element not explicitly labeled as something other than a field. Therefore, you virtually never have to explicitly label an element as a field. Note that a field element, unlike group or list, will contain information in its nodes - rather than other nodes. | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "group" | The group element is typically a structural element of the schema only, in which case it has no mapping.<br>Note that when grouping several elements that are all used to map an XML structure of a CLOB / XML field of a business object driven record, the mapping may be at the group level. | Example where a group is used to create a structure<br><br><schema><br><input type="group"<br><userId/><br></input><br><output type="group"><br><firstName/><br><lastName/><br></output><br></schema><br><br>Example where the group includes the mapping:<br><br><parameters type="group" mapXML="BO_DATA_AREA" mdField= "F1_TODOSUMEMAIL_PARM_ LBL"/><br><numberOfDays<br><br>mdField="F1_NBR_DAYS"<br>required="true"/><br><frequency<br><br>mdField="F1_FREQUENCY" /> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "list" | The list node is structural node like the group node. The only difference is that the list structure has the ability to repeat multiple times in an XML document. | Example of a schema with a list:<br><br><schema><br><statesList type="list"><br><state isPrimeKey="true"/><br><description/><br></statesList><br></schema><br><br>Example of a schema with a list:<br><br><xml><br><statesList><br><state>AK</state><br><br><description>Alaska</description><br></statesList><br><statesList><br><state>AL</state><br><br><description>Alabama</description><br></statesList><br>...<br></xml> |
| | "raw" | The raw data type is used to capture a chunk of raw text that doesn't have any inherent structure associated with it. | <sendDetail type="raw" /><br><br>Example of an XML instance for the above schema:<br><br><sendDetail><br><messageInfo><br><br><senderAddress>123 W. Main St, Ontario, CA<br></senderAddress><br><corpZone>3A</corpZone><br></messageInfo><br></sendDetail> |


Copyright © 2026, Oracle and/or its affiliates.

### The Data Type of a Field Element

Of the four different element types, only a field can have a data type.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| dataType= | "string" | By default, a field element is a string. Therefore, there is no requirement to specify the string data type. | <schema><br>  <custName dataType="string"/><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | `"number"` | Defines an element that is a number.<br><br>**Note**<br>UI hints include a setting to Suppress Automatic Number Formatting.<br><br>**Note**<br>Use **currencyRef** attribute for auto-display of currency symb | Examples<br><br>`<schema>`<br>`  <count dataType="number"/>`<br>`</schema>`<br><br>`<schema>`<br>`  <taxRate dataType="number"`<br>`   currencyRef="currency"/>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | ol that is associated with the referenced currency code. The currency decimal positions are ignored by this formatting allowing you to display a currency symbol | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | for a unit rate with many decimals. | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | `"money"`<br>Optional additional attributes<br>`currencyRef=`<br>"element name" | Defines an element that represents a monetary amount.<br>The currency reference is optional and if left blank the installation currency will be used. Automatic formatting and validation to be applied based on the currency. For example, the currency symbol will be shown when auto-rendering. In addition, the number of decimal places must not exceed the valid number defined for the currency.<br>**Note**<br>Refer to Referencing Other Elements for supported syntax for referring to other elements. | `<schema>`<br>`  <currency default="USD"`<br>`  suppress="true"/>`<br>`  <balance dataType="money"`<br>`  currencyRef="currency"/>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "lookup"<br>Required additional attribute<br>lookup=<br>"field name" | Defines an element that has valid values defined using a lookup. The lookup field is required. | <schema><br><status dataType="lookup"<br>lookup="STATUS_FLG"/><br></schema> |
| | "lookupBO"<br>Required additional attribute<br>lookupBO=<br>"bo name" | Defines an element that has valid values defined using an extendable lookup. The extendable lookup's business object is required. | <schema><br><category dataType="lookupBO"<br>lookupBO="CM-BusinessCategory"/><br></schema> |
| | "boolean" | Defines an element that has values of "Y" and "N". | <schema><br><allowsEdit dataType="boolean"/><br></schema> |
| | "date" | Defines an element that represents a date. | <schema><br><startDate dataType="date"/><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | `"dateTime"` | Defines an element that represents a date and time.<br><br>**Note**<br>Refer to Standard Time Considerations for additional configuration available for date / time fields that represent standard time. | `<schema>`<br>`  <startDateTime`<br>`  dataType="dateTime"/>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "time" | Defines an element that represents a time. | <schema><br><startTime dataType="time"/><br></schema> |
| | "uri" | Defines an element captures a URI. Elements defined with this type enable the support for URI Allowlist and Substitution as described in Referencing URIs . | <schema><br><exportDirectory dataType="uri"/><br></schema> |

### Referencing Other Elements

There are several attributes that allow for a reference to another element in the same schema. The supported syntax of the XPath reference is the same in every case. This section provides examples using the default reference attribute (**defaultRef**).

Reference a sibling element:

```
<schema>
   <id mapField="ACCT_ID" required="true"/>
   <altId defaultRef="id" required="true"/>
</schema>
```

Reference an element in a higher group:

```
<schema>
   <id mapField="ACCT_ID" required="true"/>
   <msgInfo type="group" mapXML="XML_FIELD">
     <altId defaultRef="../id" required="true"/>
   </msgInfo>
</schema>
```

Reference an element in a lower group:

```
<schema>
   <id mapField="ACCT_ID" defaultRef="msgInfo/altId" required="true"/>
   <msgInfo type="group" mapXML="XML_FIELD">
     <altId required="true"/>
   </msgInfo>
</schema>
```

Reference an element in another group:

```
<schema>
  <acctInfo type="group">
    <id mapField="ACCT_ID" required="true"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
  </acctInfo>
   <msgInfo type="group" mapXML="XML_FIELD">
     <altId defaultRef="../acctInfo/altId" required="true"/>
   </msgInfo>
</schema>
```

### Legal vs. Standard Time Considerations

Most date / time fields represent "legal" time such that if a time zone changes their clocks for winter and summer time, the date / time field captures the current observed time. However, some date / time fields should always be captured in standard time to avoid confusion / ambiguity. A good example is a date and time related to detailed interval data. Refer to Designing Time Zones for more information.

When defining an element with `dataType="dateTime"`, you may use `legalTime="true"` to indicate that data captured in the element always represents legal time in the 'base' time zone, or use `stdTime="true"` to indicate that data is stored in standard time in the 'base' time zone. The 'base' time zone is specified on the Installation options.

> **Note**
>
> If an element is mapped to a table / field with a Standard Time Type of **Physical**, then `stdTime="true"` is implied. Refer to Fields for more information.

Example:

```
<schema>
  <startTimeS dataType="dateTime" stdTime="true"/>
  <startTimeL dataType="dateTime" legalTime="true"/>
</schema>
```

If the time zone that represents the date / time field is not the installation time zone, use the optional setting `stdTimeRef`="XPath to time zone element" to indicate that the element represents standard time and indicates the time zone to use. Use `legalTimeRef`= when the element is stored in legal time. Refer to Referencing Other Elements for supported syntax for referring to other elements.

Example:

```
<schema>
  <alternateTimeZone fkRef="F1-TZONE" suppress="true"/>
  <startDateTimeS dataType="dateTime" stdTimeRef="alternateTimeZone"/>
  <startDateTimeL dataType="dateTime" legalTimeRef="alternateTimeZone"/>
</schema>
```

> **Note**
>
> If an element is mapped to a table / field with a Standard Time Type of **Referenced**, then `stdTimeRef`="XPath" is implied. Refer to Fields for more information.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When schema elements are captured in standard time the UI map supports HTML notation to automatically display the data applying a daylight savings time / summer time correction. Refer to the HTML attribute oraType="dateTime;" for more information.

There may be cases where the date / time is captured in one time zone, but should be displayed using a different time zone. In this case, the attribute `displayRef`="XPath" may be used in addition to the appropriate attribute that identifies the time zone that the data is capture in. Refer to Referencing Other Elements for supported syntax.

```
<schema>
  <displayTimeZone fkRef="F1-TZONE" suppress="true"/>
  <startDateTimeS dataType="dateTime" stdTime="true"
displayRef="displayTimeZone"/>
  <startDateTimeL dataType="dateTime" legalTime="true"
displayRef="displayTimeZone"/>
</schema>
```

### The Mapping Attributes

When constructing your schema, you can choose from one of the following mapping attributes.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| `mapField=` | "field name" | In the case of a business object, the **mapField** attribute is used to identify the database column the element is related to. For business service schemas, the `mapField=` attribute is used to link a schema element with a service element. | `<schema>`<br>`  <factId`<br><br>`mapField="FACT_ID"/>`<br>`</schema>` |
| `mapChild=` | "table name" | The **mapChild** attribute is used only for business object mapping. It is used in two ways:<br>• First, to create a list in the business object that corresponds to a child table of an MO.<br>• Second, you can use **mapChild** to identify the child table a flattened field lives in. For more information on flattening, refer to flattening section below. | Example of a list within a child table in a BO:<br><br>`<persons type="list"`<br><br>`mapChild="CI_ACCT_PER"`<br>`  <personId`<br>`mapField="PER_ID"/>`<br>`</persons>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| `mapList=` | "list name" | The **mapList** attribute is used only for business service mapping. It is used in two ways:<br>• First, to create a list in the business service that corresponds to a list in the service.<br>• Second, you can use **mapList** to identify the list that a flattened field lives in. For more information on flattening, refer to flattening section below. | Example of a list within a business service:<br><br>`<selectList type="list" mapList="DE" <value mapField="COL_VALUE"> <row mapList="DE_VAL"> <SEQNO is="2"/> </row> </value> </selectList>` |
| `mapXML=` | "field name" | The **mapXML** attribute is typically used to map XML structures into a large character / XML field of the service. Note that when you use **mapXML** to map either a list or group structure (`type="list"` or `type="group"`) you don't have to map all the child elements within the structure. It is also possible to map list elements to a large character field associated with the list child. | `<enrollmentRequest type="group" mapXML="CASE_CLOB"> <messageID/> <sender/> </enrollmentRequest> <enrollmentKey mapXML="CASE_CLOB"/> <enrollmentInfo type="list" mapChild="CI_CASE_CHILD"> <sequence mapField="CHILD_SEQ"/> <name mapXML="CASE_CHILD_CLOB"/> <value` **`mapXML="CASE_CHILD_CLOB"`**`/> </enrollmentInfo>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| isPrimeKey= | "true" | You must specify a primary key for a list defined within a mapped XML element (type="list" mapXML="CLOB"). The primary key is used by the framework to determine whether a list element add, update or delete is required during a business object update.<br><br>**Note**<br>You do not need to specify the prime key for a business object list mapped to maintenance object list. Whe | <questionnaire type="list" mapXML="CASE_CLOB"><br>  <question<br>  isPrimeKey="true"/><br>  <answer/><br></questionnaire> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | n a physical list is mapped, the primekey is derived from existing physical meta-data. | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| `orderBy=` | "XPath `asc\|desc` , XPath `asc\|desc` " | By default, a list defined within a mapped XML element (`type="list" mapXML="CLOB"`) is sorted by the first element of the list. A different sort order may be specified using the **orderBy** attribute. The attribute value is a comma separated list of field XPaths (relative to the list element) with an optional sort order (ascending is the default).<br><br>**Note**<br>This is only available for lists mapped as XML within business objects. | `<questionnaire type="list" orderBy="page/section, page/sequence" mapXML="CASE_CLOB"> <question isPrimeKey="true"/> <answer/> <page type="group"> <section/> <sequence/> </page> </questionnaire>` |

### Descriptive Attributes

The following attributes can be used to describe a schema element and provide additional configuration related to the element. Typically, these attributes are useful for field elements only.


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| `<!-- comment -->` | | Use this to add a comment to a schema by using special open and close characters: <!-- and -->. | `<schema>`<br>`  <!-- This schema is used to capture business information only, please refer to the 'HUMAN' BO for person information -->`<br>`</schema>` |
| `description=` | "text" | The description of an element may be used to provide an internal description of the element to help a reader of the schema to understand the business reason for the element. | `<schema>`<br>`  <active type="boolean" description="active account" label="Active"/>`<br>`</schema>` |
| `label=` | "text" | The label of an element is meant to be a short bit of verbiage that would typically precede the element in a user interface. | `<schema>`<br>`  <active type="boolean" description="active account"` **`label="Active"`**`/>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| `required=` | `"true"` | Used to require the existence of an element during object interaction.<br><br>**Note**<br>For included schemas, the required="true" attribute is not processed on business object and business service schemas when | `<schema>`<br>`  <logDate mapField="LOG_DT" default="%CurrentDate"`<br><br>`required="true"`<br>`private="true"`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | they are included within a service script schema. This is to support the ability of the service script to populate required elements before an embedded busin | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | ess object or business service is invoked from the service script. | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| mdField= | "field code" | The meta-data field attribute is used to associate an element with a field's metadata. The field defines data type, as well as its label and help text. If you link an element with a meta-data field, you don't need to specify any of these attributes.<br><br>**Note**<br>For a business object schema, the mapField attribute is used to derive the data type and label. An mdField attrib | <schema><br>  <active mdField="CM_ACTIVE_SW" /><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | ute may be provided to override these attributes, if needed. If the element is mapped to an XML column, the mdField is needed to provide the appropriate | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | | riate data type and label. | |
| `fkRef=` | "FK Reference Code" | If the element is a foreign key, defining its FK Reference will enable framework validation of the element during schema interaction and automatically provide descriptions and navigation capability. | `<schema> <person fkRef="PER" mapField="CHAR_VAL_FK1"> <row mapChild="CI_SA_CHAR"> <CHAR_TYPE_CD is="PER"/> </row> </person> </schema>` |
| `private=` | `"true"` | Marking an element as private will prevent it from being exposed in schema interaction.<br>**Note**<br>A private element requires a default. | `<schema> <type mdField="SA_TYPE_CD" default="E1" private="true"/> </schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| suppress= | "true" | This setting prevents an element from appearing in automatically generated user interfaces.<br><br>**Note**<br>This attribute can be specified on a group, in which case all elements of the group will be suppressed. | <schema><br>  <ls<br>mdField="LIFE_SUPPORT_FLG"<br>default="N"<br>suppress="true"/><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | | **Note**<br>Refer to Defaulting and System Variables for advice related to default values for suppressed elements. | |
| | `"blank"` | This setting means that automatic UI rendering in display mode will hide the element if its value is blank. The element will still be modifiable on the input map whether blank or not. | `<schema>`<br>`  <email mdField="EMAIL" suppress="blank"/>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | `"input"` | This setting means that automatic UI rendering will suppress the element for input, although it may still be displayed if it is not blank.<br><br>**Note**<br>Elements marked as `suppress="input"` will behave as with `suppress="blank"`. If the value is blank, no value will | `<schema>`<br>`  <email mdField="EMAIL" suppress="input"/>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | be displayed on either the display or input map. If the element's value is present, then the element will be displayed on both the display and inpu | |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |

t map.

**Note**

Refer to Defaulting and System Variables for advice related to default values for suppressed elements.


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| noAudit= | "true" | This setting prevents an element from appearing as a changed element in the business object's audit plug-in spot. If specified on a group or list node it applies to the whole node. You cannot specify it on the schema root node, only on schema elements.<br><br>**Note**<br>This attribute is only applicable to business object schemas. | <schema><br>...<br><version mapField="VERSION_NBR" noAudit="true"/><br>...<br><workFields type="group" noAudit="true"<br><lastProcessedId/><br><lastProcessedTime/><br></workFields><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| `storeEmptyNodes=` | `"true"` | By default, empty nodes are removed from a business object instance when saved. This setting allows a group or a list element to explicitly keep its empty nodes.<br>This attribute may be useful in situations where business object data is exchanged with an external system and there is a need to distinguish between an empty value for an element that participates in the synchronization and an element that does not participate and therefore omitted from the message altogether.<br>**Note**<br>This is only available for group and list elements. | `<schema>`<br>`  <message type="group" storeEmptyNodes="true">`<br>`  ...`<br>`  </message>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| `emitEmptyGroups=` | `"true"` | By default, empty nodes are not included in the output of a business object or a business service call. This attribute is set at the top level schema node, allowing empty nodes of all groups and list elements to be included in the output.<br>This attribute may be useful in the same situations that may require the use of the **storeEmptyNodes=** attribute.<br>**Note**<br>This is only available for business object and business service schemas. | `<schema emitEmptyGroups="true" >`<br>`  <message type="group">`<br>`  ...`<br>`  </message>`<br>`</schema>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| emitEmptyElements= | "true" | This attribute is similar to the **emitEmptyGroups**= attribute but may also be specified at the field level. The attribute is applicable to all schema types. | <schema><br>  <messageNumber type="field" emitEmptyElements="true"><br>  ...<br>  </message><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| adheresToDA= | “comma separated list of data area names.” | This attribute is applicable to the schema, group and list nodes. It claims that this sub-schema node is intended to "look like" the schema of one of the listed data areas, for polymorphism reasons in general.<br><br>**Note**<br>The recommended approach for enforcing the structure of a shared data area is by including it in the schema. Thi | <schema><br><messageNumber type="group" adheresToDA="DataAreaName"><br>...<br></message><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
| --- | --- | --- | --- |
| | | s attribute should be limited to use cases where the ability to include the data area is not possible. | |

### Schema Constants

There are some product owned schemas where the design warrants a value to be defaulted in the schema, but where the value is implementation specific and therefore cannot be defined by the product. For these situations, the product may use a technique called a schema constant. The design of the schema will include a declared constant. At implementation time, a configuration task will include defining the appropriate value for the constant.

For example, imagine the product delivers an algorithm that will create an outbound message when a certain condition occurs. The outbound message type to use must be configured by the implementation. To use a schema constant to define the outbound message type, the base product will configure the following:

- An option type lookup value for the lookup `F1CN_OPT_TYP_FLG` is defined. For example, `M202` - Activity Completion Outbound Message Type with a Java Value Name of **outmsgCompletion**


Copyright © 2026, Oracle and/or its affiliates.

- The base schema that is used to create the "complete activity" outbound message references the schema constant using the Java Value Name of the option type's lookup value

```
...
<outboundMessageType mapField="OUTMSG_TYPE_CD"
default="%Constant(outmsgCompletion)"/>
...
```

At implementation time, the administrative users must configure the appropriate outbound message type for "activity completion". Then, navigate to Feature Configuration, choose the **Schema Constants** feature type, choose the option type **Activity Completion Outbound Message Type** and enter the newly created outbound message type in the option value.

Schema constants may also be used in the flattening syntax to define the row elements required for flattening.

#### Defaulting and System Variables

The default attribute can be used to default values into field elements as well as the row elements required for flattening . You can default a field to a constant or to one of several system variables.

When an element is required, the default value is applied at the server level when adding or changing the record where no value is provided for this element.

When an element is optional, a default value may be configured to provide a suggested value to the user when populating the record on the user interface in Add mode. Users should be able to remove the default value for this type of element and the system will not attempt to populate it at save time.

> **Note**
>
> The standard for our 'list' maintenance is to show an empty row for a new list that a user may or may not choose to populate. As such, UI default values are not shown for list elements.
>
> It is not recommended to configure a default attribute for an optional element that is not editable or visible on the user interface in Add mode (for example an element with `suppress="true"` or `suppress="input"` or a list element whose default value is not shown per the previous point). The default for this type of element is still applied but the user doesn't have the ability to reset the value. This may result in inconsistencies as defaulting does not occur for optional fields added by a use case other than the UI interaction.


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| default= | "value" | Use this attribute to default an element to a specified value. The values that are valid depend on the **dataType** setting. | <schema><br><perType<br>mapField="PER_OR_BUS_FLG"<br>default="P"<br>required="true"/><br></schema> |
| | | | <schema><br><frequency<br>dataType="number"<br>default="1"<br>required="true"/><br></schema> |
| | "%CurrentDate" | Used to default the element to the current date. This is only applicable to date elements. | <schema><br><logDate<br>mapField="LOG_DT"<br>default="%CurrentDate"<br>required="true"/><br></schema> |
| | "%CurrentDateTime" | Used to default the element to the current date / time. This is only applicable to date / time elements. | <schema><br><logDateTime<br>mapField="LOG_DTTM"<br>default="%CurrentDateTime"<br>required="true"/><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "%StandardDateTime" | Used to default the standard date and time. The standard date and time is identical to the current date and time, unless daylight savings time / summer time is in effect for the base time zone. This may be used with the **stdTime** attribute.<br><br>**Note**<br>Refer to Standard Time Considerations for more information. | <schema><br><startDateTime mapField="START_DTTM" default="%StandardDateTime" required="true"/><br></schema> |
| | "%ProcessDate" | You can default the process date. The process date differs from the current date because the process date will remain constant throughout the duration of the process being executed. The current date will reflect the actual date of processing. This is similar to the batch business date that is a standard batch parameter. | <schema><br><billDate mapField="BILL_DT" default="%ProcessDate"<br><br>required="true"/><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "%ProcessDateTime" | This is similar to "%ProcessDate" but for date / time fields. | <schema><br><calcDateTime mapField="CALC_DTTM" default="%ProcessDateTime" required="true"/><br></schema> |
| | "%CurrentUser" | Used to default the element to the current user. | <schema><br><logUser mapField="LOG_USER" default="%CurrentUser" required="true"/><br></schema> |
| | "%CurrentUserTimeZone" | Used to default the element to the current user's time zone. If the current user's time zone is not found, the installation time zone is used. | <schema><br><timeZone default="%CurrentUserTimeZone" required="true"/><br></schema> |
| | "%CurrentUserLanguage" | Used to default the element to the current user's language. | <schema><br><custLanguage mapField="CUST_LANG" default="%CurrentUserLanguage" required="true"/><br></schema> |
| | "%InstallationCurrency" | Used to default the currency from the installation record. | <schema><br><currency mapField="CURRENCY_CODE" **default="%InstallationCurrency"** required="true"/><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "%InstallationCountry" | You can default the country from installation record. | <schema><br>  <country<br>mapField="COUNTRY"<br>default="%Installation Country"<br>required="true"/><br></schema> |
| | "%InstallationLanguage" | You can default the language from the installation record. | <schema><br>  <language<br>mapField="LANGUAGE"<br>**default="%Installation Language"**<br>required="true"/><br></schema> |
| | "%Constant( )" | You can default an element value using a schema constant. | The following is an example of a schema constant used as a default value, where the Java Value Name of the Lookup Value is 'customerLanguage'.<br><br><language<br>mapField="CUSTOMER_LANG"<br>default="%Constant(customerLanguage)"<br>required="true"/> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | `"%Context( )"` | You can default a value contained in a context variable. | An example of a context variable used as a default value:<br>`<source mapField="PER_ID" **default="%Context(source)"** required="true"/>` |

**Warning**

Context variables must be initialized within a server script before the schema context default can be applied. Refer to Conte

**Note**

When defining a context variable in scripting, it should be prefixed with $$. When referring to the varia


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | | xt Variables for more information. | ble in the **%Context()** syntax, the prefix is not included. |
| `defaultRef=` | "XPath" | Use this attribute to default the value of one element to the value of another one. | Refer to Referencing Other Elements for supported syntax for referring to other elements. |

### The Flattening Nodes and Attributes

The term "flattening" is used to describe the act of defining one or more single elements for a schema that are actually part of a list within the maintenance object. Flattening is possible if there are other attributes of the list that can be defined to uniquely describe the element or elements. A common use case for flattening is a characteristic. Rather than defining the characteristics of an object using a collection where the user must choose the characteristic type and then define the value, the characteristics are defined as actual elements with the appropriate label already displayed. This technique enables the designer of the schema and the user interface to display each separate characteristic in the logical place in the user interface rather than all lumped together.

> **Note**
>
> A flattened element represents a unique row in the database. This row is inserted when the flattened values are created. The row is updated when any of the flattened values are changed. The row is deleted when all the flattened values are removed. The behavior of effective dated elements is slightly different - please see Flattening an Effective Dated List.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The flattening feature can also be used to define a list, see Flattened List.

### Identifying the List or Child Table

When flattening a child table, the row node is required to identify the list / child table that the element comes from. Within the row node, at least one element must be defined with an `is=` definition that ensures that the element is a unique row in the database. It may also define elements or fields in the row that are suppressed and are populated using default value configuration.

- For a business object, the row node defines the child table the flattened field belongs to.

  The syntax is **<row mapChild="table name">**. This example is for the list of persons for an account in the customer care and billing product. One person may be marked as the "main" person. This illustrates how to define an explicit element for the main person ID to simplify references to that field. It is part of the CI_ACCT_PER child table. What makes it unique is that the MAIN_CUST_SW is **true** (and only one row may have that value)

  ```
  <custId mapField="PER_ID" mdField="CM-MainCust">
    <row mapChild="CI_ACCT_PER">
      <MAIN_CUST_SW is="true" />
      <ACCT_REL_TYPE_CD default="MAIN" />
    </row>
  </custId>
  ```

  > **Note**
  >
  > The above example illustrates that the row node may also define elements within the list that are suppressed and assigned a default value. This syntax is never used to identify a particular row. Note that a default value can either be a literal string, or a system variable.

- For a business service, the row node identifies the list name the flattened field belongs to.

  The syntax is **<row mapList="list name">**. This example shows two entries from a list being flattened to a field value and description.

  ```
  <selectList type="list" mapList="DE">
      <value mapField="COL_VALUE">
      <row mapList="DE_VAL">
            <SEQNO is="2"/>
      </row>
      </value>
    <description mapField="COL_VALUE">
      <row mapList="DE_VAL">
          <SEQNO is="3"/>
      </row>
    </description>
  </selectList>
  ```

### Uniquely Identifying the Flattened Field or List

The `is=` syntax within a row or rowFilter element is used to uniquely identify the row.


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| is= | "value" | Use this attribute to reference a value directly. | <tdTypeCd<br>mapField="CHAR_VAL_FK1"><br><row<br><br>mapChild="F1_EXT_LOOKUP_VAL_CHAR"><br><CHAR_TYPE_CD<br>**is="CM-TD-TYPE"**/><br></row><br></tdTypeCd> |
| | "%Constant( )" | You can configure a flattened element using a schema constant. During a service interaction the value of the schema constant will be used to identify the flattened element in its child row. | An example of a schema constant used in flattening syntax where the Java Value Name of the Lookup field value is 'cmRate'.<br><br><unitRate mapField="CHAR_VAL"<br>dataType="number"><br><row mapChild=<br>"F1_EXT_LOOKUP_VAL_CHAR"><br><CHAR_TYPE is<br>**="%Constant(cmRate)"**/><br></row><br></unitRate> |
| | "%n" | The %n substitution value is used to reference a relative list instance. A relative list instance is typically used to configure a flattened element for a child table keyed by sequence number. The value of *n* should be a positive integer value. During a business object read interaction the relative list instance (position) specified by the integer will be returned. | An example with a relative list instance - where the **first** instance of the row is returned.<br><br><schema><br><mainPhone mapField="PHONE"><br><row<br>mapChild="CI_PER_PHONE"><br><PHONE_TYPE_CD<br><br>is="%Constant(mainPhoneType)"/><br><SEQ_NUM**is="%1"**/><br></row><br></mainPhone><br></schema> |

**NOT_SUPPORTED**

Additional values for is= are used when Flattening an Effective Dated List. Refer to that section for more information.

### Flattening a Pre-defined Characteristic Type


Copyright © 2026, Oracle and/or its affiliates.

If the flattened field is in a characteristic collection and the characteristic type is a predefined characteristic, automatic UI rendering will generate a dropdown for the list of valid values. For example, the schema below will generate a dropdown for the Usage element showing the valid values of the Status Reason Usage (F1-SRUSG) characteristic type.

```
<usage mdField="STATUS_RSN_USAGE" mapField="CHAR_VAL">
  <row mapChild="F1_BUS_OBJ_STATUS_RSN_CHAR">
      <CHAR_TYPE_CD is="F1-SRUSG"/>
      <SEQ_NUM is="1"/>
  </row>
</usage>
```

### Defining Multiple Elements from the List

When attempting to include multiple columns from the same list, the system provide shorthand notation for copying the flattening rules defined on another element so that the flattening rules do not need to be repeated. To do this, the row node includes the **rowRef** attribute and it indicates the other element name that defines the mapping information. The following example illustrates flattening the fields Customer ID and Receives Copy of Bill from the same list of Persons for an Account (where the MAIN_CUST_SW is **true** ).

```
<custId mapField="PER_ID">
  <row mapChild="CI_ACCT_PER">
      <MAIN_CUST_SW is="true" />
      <ACCT_REL_TYPE_CD default="MAIN" />
  </row>
</custId>
<copyBill mapField="RECEIVE_COPY_SW" rowRef="custId"/>
```

Note that the above notation also illustrates that the **rowRef** may be defined directly in the element's attribute definition.

> **Note**
>
> Refer to Referencing Other Elements for supported syntax for referring to other elements.

### Flattening Two Layers Deep

If your maintenance object or service has nested lists two layers deep, the system supports flattening an element within a flattened element. This technique also uses the **rowRef** attribute. The flattening of the second level refers to the flattened element of the first level. The following example illustrates flattening a characteristic into an element called legalContact for the "main" customer. Notice that the legalContact element has two row nodes: one to refer to the flattening information for its parent record and one to define its child table

```
<custId mapField="PER_ID">
  <row mapChild="CI_ACCT_PER">
    <MAIN_CUST_SW is="true" />
    <ACCT_REL_TYPE_CD default="MAIN" />
  </row>
</custId>
<legalContact mapField="CHAR_VAL_FK1">
  <row rowRef="custId">
```


Copyright © 2026, Oracle and/or its affiliates.

```
    <row mapChild="CI_ACCT_PER_CHAR" >
       <CHAR_TYPE_CD is="LEGAL" />
    </row>
  </row>
</legalContact>
```

Note that the above notation also illustrates that the **rowRef** may be defined as an attribute of a row node rather than directly in the element's attribute definition.

**Defining a Flattened List**

There are times that a list or child table supports multiple values of the same "type" and these should be presented as a list. To continue with the example above, the list of persons for an account may identify one person as the "main" person. This person has been flattened to a single element (with the account relationship type defaulted and suppressed). To maintain the other persons related to an account, you can define a list where each row captures the Person Id and the Account Relationship Type.

Rather than a row node, the flattened list is configured with a **rowFilter** element. The following schema illustrates the described example. The list node defines the child table. The **rowFilter** includes the criteria that identify the rows within the table to include. The elements of the list are defined within the list node outside the **rowFilter** element.

```
<custId mapField="PER_ID">
  <row mapChild="CI_ACCT_PER">
    <MAIN_CUST_SW is="true" />
    <ACCT_REL_TYPE_CD default="MAIN" />
  </row>
</custId>
<miscPersons type="list" mapChild="CI_ACCT_PER">
  <rowFilter>
    <MAIN_CUST_SW is="false" />
  </rowFilter>
  <relType mapField="ACCT_REL_TYPE_CD"/>
  <personId mapField="PER_ID"/>
</custId>
```

Note that the system will validate that if a schema contains flattened single elements and flattened lists from the same child table, the criteria that defines what makes them unique must be analogous.

**Flattening an Effective Dated List**

There are some lists in the application that are effective dated (and still others that have effective date and time). For example, there are some effective dated characteristic collections. In these collections, the design is to capture a single value for a characteristic type that may change over time. It is not meant to support multiple characteristic values in effect at the same time.

There are some maintenance objects that have effective dated characteristics and no sequence based characteristics. For those entities, implementations may want to use the characteristic to define an additional field that doesn't require effective dates.

When considering designing an element for your schema that represents a value in an effective dated collection, consider if the value is one that is used in processes, like calculating a bill, such that it is important to capture changes to the value over time.


Copyright © 2026, Oracle and/or its affiliates.

- If it is not important to track the changes to the value, the recommendation is to set the effective date to a fixed value in the schema. (Examples below).
- If it is important to track the changes to the value, then you need to consider how you want to design the schema.
  - One option is to define a flattened list as described above. Both the value and the effective date are elements in this list. In this use case, you manage the list just like any list. You can add or remove values.
  - Another option is to define a flattened single value in the schema. With this option, when retrieving the record, the latest effective value is returned. In this design, the important question is how is the reference date defined.

> **Note**
>
> Refer to Referencing Other Elements for supported syntax for referring to other elements.

### Flattened Elements Where Changes Are Not Tracked

The following is an example of flattening an element in an effective dated collection where the value does not need to be effective dated. The recommendation in this case is to simply hard-code the date in the `is=` attribute.

```
<schema>
  <externalReference mapField="ADHOC_CHAR_VAL" mdField="CM_EXT_REF"
dataType="string">
    <row mapChild="CI_SA_CHAR">
      <CHAR_TYPE is="CM_EXT_REF"/>
      <EFFDT is="2000-01-01"/>
    </row>
  </externalReference>
</schema>
```

### Flattened Elements Where Changes Are Effective Dated

If you want the BO schema to flatten an effective dated value that should keep track of changes over time, the flattened value will represent the latest effective entry. In this case, the recommendation is to explicitly define the date or date/time field as an element. Then there is syntax for the `is=` attribute to indicate that the explicit date field is the date to use for the value.


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| is= | "%effectiveDate( )" | Use this configuration to indicate that the date to use is the value of another element.<br><br>**Note**<br>Refer to Referencing Other Elements for supported syntax for referring to other elements. | <schema><br><price mapField="CHAR_VAL"<br>dataType="number"<br>required="true"><br><row mapChild="CI_SA_CHAR"><br><CHAR_TYPE is="PRICE"/><br><EFFDT<br>is="%effectiveDate(priceEdate)"<br>/><br></row><br></price><br><priceEdate mapField="EFFDT"<br>rowRef="price"/><br></schema> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | `"%effectiveDateTime()"` | Use this configuration to indicate that the date / time to use is the value of another element.<br><br>**Note**<br>Refer to Referencing Other Elements for supported syntax for referring to other elements. | `<schema>`<br>`  <price mapField="CHAR_VAL"`<br>`  dataType="number">`<br>`  <row mapChild="RATE_CHAR">`<br>`  <CHAR_TYPE is="PRICE"/>`<br>`  <EFFDTTM`<br><br>`is="%effectiveDateTime(priceEdatetime)"/>`<br>`  </row>`<br>`  </price>`<br>`  <priceEdatetime mapField="EFFDTTM"`<br>`  rowRef="price"/>`<br>`</schema>` |

You should consider if the explicit date or date/time element should be displayed on the user interface when the flattened field is shown or if that should be suppressed. Designers should also consider the user dialogue for updating the value. It may be more intuitive for the flattened field to be display only and to have a separate dialogue for updating the effective dated value.

For this option, the suggestion is that the maintenance page for the object includes a separate zone that shows the history of the effective dated values over time. The latest value may be included in that zone as well.

**Note** that for backward compatibility, the system also supports the following options. However, they are not recommended. They implicitly defined the reference date as the current date at all times. The options supported adding new effective dated rows when the value changed. But the technique also caused new effective dated rows to appear when other elements of the same schema were updated. The new rows would have the current date or date/time with the same row value because the reference date or date/time is the current date, which is a


Copyright © 2026, Oracle and/or its affiliates.

change. The product does not recommend managing an effective dated characteristic without defining the date or date/time as an explicit element.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| is= | "%effectiveDate" | This configuration is not recommended | <schema><br><price mapField="CHAR_VAL"<br>dataType="number"><br><row<br>mapChild="CI_SA_CHAR"><br><CHAR_TYPE is="PRICE"/><br><EFFDT<br>is="%effectiveDate"/><br></row><br></price><br></schema> |
| | "%effectiveDateTime" | This configuration is not recommended | <schema><br><price mapField="CHAR_VAL"<br>dataType="number"><br><row mapChild="RATE_CHAR"><br><CHAR_TYPE is="PRICE"/><br><EFFDT<br>is="%effectiveDateTime" /><br></row><br></price><br></schema> |

### Search Zone

A UI Map schema element can be configured to enable an automatic search dialog when the schema is included within a maintenance UI map.

> **Note**
>
> Please note that an fkRef can be configured with a search zone. If a schema element has an fkRef but no explicit search attributes (as described here) then the fkRef search information will be used in the UI map. In other words, if the schema element already has an fkRef, then these explicit search attributes in the schema are only used to override the fkRef search information.

> **Note**
>
> Refer to the UI Map Attributes and Functions for more information on search zone configuration.

**search="search zone"**


Copyright © 2026, Oracle and/or its affiliates.

The search attribute can be used within a UI map schema and is used to automatically generate the oraSearch UI map attribute. The search zone is an explorer zone configured as a search.

```
<person fkRef="PER" search="C1_PSRCH"/>
```

**searchField="search field:element|'literal';"**

The searchField attribute can only be used in conjunction with the search attribute. The searchField attribute is used to build the oraSearchField UI map attribute. The searchField value is used to populate a search zone filter with an initial value when the search zone is launched. The initial value can be a literal also. The searchField value is used to match to the filter mnemonic also named searchField.

Search field: element|'literal'. The search field represents the search zone filter to populate on launch. The element is the map's schema element used to populate the filter. The element is optional, if left blank, it will default to the element that this attribute is specified on. The searchField also takes 'literal' as input value

> **Note**
>
> Multiple filters can be populated within the search zone at launch, but multiple search field pairs must be constructed within the attribute value. The value specified here will be used to directly build the HTML attribute oraSearchField within the UI map where this schema is specified.

> **Note**
>
> Note that the element reference is *relative*. Refer to Referencing Other Elements for supported syntax for referring to other elements.

```
<person fkRef="PER" search="C1_PSRCH" searchField="PERSON;
PER_TYPE:personType;"/>
```

**searchOut="search field:element;"**

The searchOut attribute can only be used in conjunction with the search attribute. The searchOut attribute is used to build the oraSearchOut UI map attribute. The searchOut value is used to capture a selected value from the search zone and move it to a UI map element. The searchOut value specified should match the ELEMENT_NAME mnemonic within the search result zone parameter.

Search field: element. The search field represents the search zone result brought back to the UI map. The element is the map's schema element to be populated. The element is optional, if left blank, it will default to the element that this attribute is specified on.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Multiple elements can be populated as a result of search zone selection, but multiple search field pairs must be constructed within the attribute value. The value specified here will be used to directly build the HTML attribute oraSearchOut within the UI map where this schema is specified.

> **Note**
>
> Note that the element reference is *relative*. Refer to Referencing Other Elements for supported syntax for referring to other elements.

```
<person fkRef="PER" search="C1_PSRCH" searchField="PER_ID"
searchOut="PER_ID;PRIMARY_PHONE:personPhone;"/>
```

### Extend Security for Service Script

Application service security will be enforced when either a business object or a service script is invoked from a BPA script or a UI map, but not from a service script. If you want security to be enforced when the business object or a service script is invoked from a service script, you must add the following attribute to the service script's schema.

**appSecurity="true"**

The appSecurity attribute is only available for service script schemas. If specified, any business object or service script directly invoked by the service script will have their application service evaluated for access.

```
<schema appSecurity="true">
  ...
</schema>
```

### Overriding Action for a Business Service

If you want to invoke a business service with an action other than 'read', you need to specify the action attribute on the primary node business service schema.

**pageAction="add, change, delete"**

The action attribute is used to override the default action of read on a business service schema. Valid values are:

- add
- update (only allowed for maintenance object service)
- change (not allowed for maintenance object service)
- delete
- read (this is the default action if no pageAction specified)

Example:

```
<schema pageAction="change">
  <parm type="group">
```


Copyright © 2026, Oracle and/or its affiliates.

```
    <ele1/>
    <ele2/>
  </parm>
</schema>
```

### Specifying searchBy for a Search Service

If you want to invoke a search service then you must explicitly specify the searchBy attribute appropriate for the elements mapped in the schema.

**searchBy="MAIN"**

The value values of the searchBy attribute can be found by viewing the XML schema linked to the business service, use the View XML URL. Typical values are:

- MAIN
- ALT
- ALT2
- ALT3
- etc.

```
<schema searchBy="MAIN">
  <AccountID mapField="ACCT_ID"/>
  <Results type="list">
    <AccountID mapField="ACCT_ID"/>
  </Results>
</schema>
```

### Including Other Schemas

There are no limitations on your ability to include a schema into another schema - all types can be included in all other types. Nested includes are also allowed - and at present there is no limitation on the depth of the nesting.

Including a schema requires two parts:

1. The include node
2. The name attribute

The following table highlights the supported include statements.


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description | Examples |
|---|---|---|
| <includeBO name=" "/> | Including a business object schema into another schema is allowed.<br>Note that the mapping rules of a business object or business service schema may or may not make sense in the context of the parent schema. Include other schemas at your own risk. However, a very useful aspect of XML processing is that the framework ignores non-pertinent attributes. In other words, it will not hurt to have mapping attributes included into a script schema. | <schema><br><cust type="group"><br><includeBO name="C1-Person"/><br></cust><br><spouse type="group"><br><includeBO name="C1-Person"/><br></spouse><br></schema> |
| <includeBS name=" "/> | Used to include a business service schema. | <schema><br><includeBS name="F1-ReadMOLog"/><br></schema> |
| <includeDA name=" "/> | Used to include a data area schema. | <schema><br><includeDA name="F1CommonSchemaFieldData"/><br></schema> |
| <includeMP name=" "/> | Used to include a UI map schema. | <schema><br><includeMP name="F1-DisplayRecordActions"/><br></schema> |
| <includeSS name=" "/> | Used to include a service script schema. | <schema><br><includeSS name="F1-ActShowZn"/><br></schema> |

### Compatibility Attributes

These attributes were added as part of upgrades from previous versions of the Framework.

**fwRel="2"**

This attribute has been added to schemas created in Framework 2 as part of a Framework 4 upgrade. New schemas will not need this attribute. It is not advisable to modify this attribute without understanding the following behavior differences as improper changes could result in errors:


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Schemas created in Framework 2 with the fwRel="2" attribute will store any XML mapped fields under groups as top-level XML elements in the mapXML field. This means that if two or more fields, in different group structures, were to have the same field name, their storage would conflict with one another resulting in errors. The new behavior, without the fwRel="2" attribute, will preserve the group structure and avoid the conflicts.

```
<schema fwRel="2"
    ...
</schema>
```

# Schema Designer

The Schema Designer is a user-friendly interface for performing the following common schema editing tasks:

- Displaying existing schemas.
- Creating schema elements.
- Moving elements within a schema.
- Adding attribute values.

The designer provides two view modes:

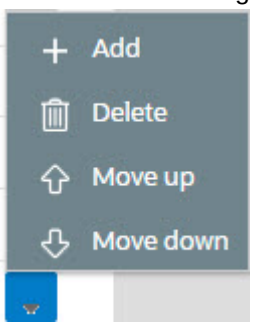
- **Text** mode shows the schema elements and their attributes written in the proper syntax and allows for direct text entry.
- **Tree** mode is a view showing the elements in a tree format with many of the key attributes of each element in tabular form. While in this mode, the **Node Display** controls allow the user to choose between displaying the elements by their internal identifiers or their associated screen labels while viewing or editing the schema.

> **Note**
>
> Schemas that are not owned by the current installation owner are protected in both view modes.

The following sections provide more information about functionality available in the Tree mode.

- If the schema definition refers to another schema using an 'include' statement, a triangle is visible to the left of the tree area. One can expand that schema by clicking the triangle.
- Detailed information about an element's definition can be displayed and if applicable, updated from the**Tree** view by clicking the edit icon for that element, which appears on the right. Note that the element attributes are only editable if the element is defined in this schema. When viewing the attributes of an element that is part of an 'include' of another schema, the attributes are display only.
- Context-sensitive embedded help is provided for fields and controls in the edit pane by clicking the Help icon


Copyright © 2026, Oracle and/or its affiliates.

- There is a menu dropdown icon visible to the left of the tree for any element that is defined in the current schema being viewed. It is not visible for elements included from another schema. New elements may be added in the **Tree** mode. Clicking that dropdown displays options to Add, Delete or Move an element. 

Note that not all options are visible, based on what is allowed for the element adjacent to the menu. For example, on the main "schema" node, only the Add option is visible as deleting or moving the "schema" node is not applicable.

When adding a new element, you are prompted first for the element position, which is either a sibling node to the current element or a child node of that element. For either option, the new element is added below the one adjacent to where the menu is clicked. You are then prompted for the element type.

The following lists the possible element types. Most are self explanatory and represent standard schema options.

- Characteristic List. This is a special type of Flattened List element that is used to map list elements that include a sequence and a characteristic value for a given characteristic type. This element is only applicable for maintenance objects that have one or more characteristic child tables where the primary key is the maintenance object's key, characteristic type and sequence. Effective dated characteristic collections are not supported. The user defines the list name, the characteristic value element name and the characteristic type. The system will configure the remaining flattening information accordingly.
- Characteristic. This is a special type of Flattened Field element that is used to map a single element to the characteristic for a given characteristic type. This element is only applicable for maintenance objects that have one or more characteristic child tables where the primary key is the maintenance object's key, characteristic type and sequence. Effective dated characteristic collections are not supported. The user defines the element name and the characteristic type. The system will configure the remaining flattening information accordingly.
- Comment. This adds a comment to the schema.
- Embedded HTML. This is specific to a schema enabled for UI Hints. It is used to include a UI map fragment.
- Field
- Flattened Field
- Flattened List
- Group
- Include BO Schema
- Include BS Schema
- Include DA Schema
- Include Map Schema


Copyright © 2026, Oracle and/or its affiliates.

- Include SS Schema
- Input Map Title. This is specific to a schema enabled for UI Hints. It is used to define a title element for the map.
- List
- Nested Flattened Field. This is a flattened field from a child table.
- Raw Element. This element is used to capture text as is. It is typically used to capture an XML structure without any details of the definition of the individual nodes.
- Section. This is specific to a schema enabled for UI Hints. It is used to define a section within the map.
- Simple Field. This is a special type of Field element that is used to define an element that is mapped to a column that supports data defined in an XML structure. (This is either a column with the character large object data type (CLOB) or the XML data type).

The **Schema Designer** is available by choosing the **Schema** tab on the Business Object, Data Area, UI Map, Business Service, and Script pages.

## Schema Viewer

The schema viewer shows a tree-view presentation of a schema in its expanded form.

> **Note**
>
> If the schema includes presentation instructions and attributes they are not included in this view. Refer to Understanding UI Hints for more information.

The schema illustrates the structure to be used when communicating with the schema's associated object. The following takes place when a schema is expanded:

- If the schema definition includes references to other schemas, these references are replaced with the corresponding schema definitions.
- Also, if the schema definition contains **private** elements, they are omitted from this view.

Clicking on any node on the tree populates the text box on the top with the node's absolute XPath expression. You will find this feature very useful when writing scripts interacting with schema-based objects. Scripting often involves referencing elements in a schema-based XML document using their absolute XPath expression. You can use this feature on the schema viewer to obtain the XPath expression for an element and copy it over to your script.

# Business Event Log

Business Event Log may be viewed as a tool designed to capture any type of business event worth noting. You configure business objects to represent the various types of events your application calls for. The following type of details may be captured for each event:

- The business object representing the type of event.
- The date and time the event took place and who initiated it.
- The business entity for which this event is logged.
- Standard application message to describe the event.


Copyright © 2026, Oracle and/or its affiliates.

- Additional context information that is available at the time of the event and varies for each type of event. The Business Event Log maintenance object supports a standard characteristics collection as well as an XML storage (CLOB) field. The event's business object determines where each piece of information resides. Refer to Business Objects for more information.

One common type of event may be the audit of changes made to sensitive data, for example, tracking an address change. Whenever an entity associated with a business object is added, changed, or deleted the system summarizes the list of changes that took place in that transaction and hands them over to **Audit** business object algorithms to process. You may design such an algorithm to audit the changes as business event logs. Refer to a business object may define business rules for more information.

You can also allow users to initiate business event logs to capture important notes about a business entity by exposing a BPA Script to invoke the event's corresponding business object.

Bottom line is that any process can create a business event log by invoking the business object representing the appropriate type of event.

# Miscellaneous Topics

The following sections describe miscellaneous system wide topics.

- Javadocs Viewer
- Module Configuration
- Global Context Overview
- System Data Naming Convention
- Accessibility Considerations
- Referencing URIs
- Caching Overview
- Expression Parser
- Debug Mode
- Prepare Issue Details
- System Override Date
- Additional Search Options

## Javadocs Viewer

The system provides a Javadocs viewer and a Groovy Javadocs viewer, which are interactive tools that allows you to browse Java documentation files (Javadocs) for Java classes that execute on the application server. For information about how to use Javadocs and navigate within the viewer, refer to Javadocs help.

### Javadocs

To launch the Javadocs viewer, open open **Admin**, and then **Implementation Tools**, and then **Javadocs**.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Proprietary Java Classes. A small number of Java classes have been suppressed due to their proprietary nature.

Additionally, when viewing a record that references a Java program in the algorithm type page, the batch control page or the FK reference page, an icon adjacent to Java program allows you to view the Javadocs for that program.

#### Groovy Javadocs

The Groovy Javadocs viewer includes the Java classes that are accessible to Groovy code within scripts. Launch this viewer using the 'View Groovy Javadocs' action on the script page.

## Module Configuration

The system provides the ability to simplify the user interface based on functionality areas practiced by your organization.

Menu items and other user interface elements are associated with function modules. By default, all function modules are accessible. If a function module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information on how to turn off a module.

If a function module is made non-accessible, i.e. turned off, its related elements are suppressed from the user interface. In addition the system may validate that related functionality is not accessed. This also means that turning off the wrong module may cause any of the following to occur:

- Menu items may not appear. Refer to Menu Item Suppression to better understand how menu item suppression works.
- Entire menus may not appear. Refer to Menu Suppression to better understand how menu suppression works.
- Tabs on pages may not appear.
- Fields may not appear.
- The system may return an error message when you attempt to use a function (indicating the function is turned off).

To correct the above situation, simply remove the module from the turned off list thus making it accessible again.

Your module configuration setup is displayed on the installations record.

- Menu Item Suppression
- Menu Suppression
- Turn Off A Function Module

## Menu Item Suppression

The following points describe how your module configuration can suppress menu items.

- Menu items that are owned by the base product (as opposed to those your implementation adds) are associated with one or more function modules. If your module configuration has


Copyright © 2026, Oracle and/or its affiliates.

turned off all of the menu item's modules, the menu item is suppressed. If at least one of the modules is accessible, i.e. turned on, the menu item is not suppressed.

- If a menu line doesn't contain any accessible items, the menu line is suppressed.
- If all lines on the Admin Menu are suppressed, the menu itself is suppressed. This is not expected to ever occur for the Main menu as the user preferences is on that menu and the expectation is that all users have access to their preferences.
- Menu items that are suppressed are not visible in the Menu Item Search results.

## Menu Suppression

In addition to the above Menu Item Suppression logic, the following points describe how your module configuration can suppress an entire menu.

- Menus that are owned by the base product (as opposed to those your implementation adds) are associated with one or more function modules.
- If your module configuration has turned off all of the menu's modules, the entire menu is suppressed. If at least one of the modules is accessible, i.e. turned on, the menu is not suppressed.

## Turn Off A Function Module

The base package is provided with a **Module Configuration** Feature Configuration that allows your organization to turn off base package function modules.

To turn off any of the base package function modules add a **Turned Off** option to this feature configuration referencing that module. Refer to the `MODULE_FLG` lookup field for the complete list of the application's function modules.

Any module not referenced on this feature configuration is considered turned on, i.e. accessible. To turn on a module, simply remove its corresponding **Turned Off** option from this feature configuration.

You may view your module configuration setup on the installation options page.

> **Note**
>
> **Only one**. The system expects only one **Module Configuration** feature configuration to be defined.

# Global Context Overview

The framework web application provides each product the ability to nominate certain fields to act as a "global context" within the web application. For example, in Oracle Utilities Customer Care and Billing, the global context fields include Account ID, Person ID and Premise ID. The values of these fields may be populated as a result of searching or displaying objects that use these fields in their keys. If you navigate to the Bill page and display a bill, the global context is refreshed with the Account ID associated with that bill. The global context for Person ID and Premise ID are refreshed with data associated with that account.

The fields designated as global context for the product are defined using the lookup `F1_UI_CTXT_FLDS_FLG`.


Copyright © 2026, Oracle and/or its affiliates.

Changing the values of the global context typically cause data displayed in zones on the sidebar to be refreshed to show information relevant to the current values of these global context fields.

When the value of one of the global context fields changes, an algorithm plugged into the installation record is responsible for populating the remaining global context values accordingly. Refer to your specific product for more information about the base algorithm that is provided for that product.

# System Data Naming Convention

There are several maintenance objects in the system that include owner flag in one or more of its tables. We refer to the data in these tables as "system data". Some examples of system data tables include Algorithm Type, Batch Control, Business Object and Script. Implementations may introduce records to the same tables. The owner flag for records created by an implementation is set to **CM** (for customer modification), however the owner flag is not part of the primary key for any of the system data tables. As a result, the base product provides the following guidelines for defining the primary key in system data tables to avoid any naming conflict.

- Base Product System Data
- Implementation System Data

## Base Product System Data

For any table that includes the owner flag, the base product will follow a naming convention for any new data that is owned by the base product. The primary key for records introduced by the product is prefixed with `xn-` where **xn** is the value of the owner flag. For example, if a new background process is introduced to the framework product, the batch code name is prefixed with `F1-`.

> **Note**
>
> There are some cases where the hyphen is not included. For example, portal codes omit the hyphen.

For most system data, the remainder of the primary key is all in capital case. An exception is schema oriented records. For business objects, business services, scripts, data areas and UI maps, the product follows the general rule of using CapitalCase after the product owner prefix. For example, `F1-AddToDoEntry` is the name of a base product business service.

> **Note**
>
> **Data Explorer Business Services.** For business services used to invoke a data explorer zone, it is recommended to name the Business Service the same name as the related zone rather than defining a different CapitalCase name for the business service.

Please note that this standard is followed for all new records introduced by the base product. However, there are base product entries in many of these system data tables that were introduced before the naming convention was adopted. That data does not follow the naming convention described above.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Schema naming conventions.** A context sensitive "Schema Tips" zone is associated with any page where a schema may be defined. The zone provides recommended naming conventions for elements within a schema along with a complete list of the XML nodes and attributes available to you when you construct a schema.

## Implementation System Data

When new system data is introduced for your implementation you must consider the naming convention for the primary key. The product recommends prefixing records with **CM**, which is the value of the owner flag in your environment. This is consistent with the base product naming convention. This convention allows your implementation to use the CM packaging tool in the Software Development Kit as delivered. The extract file provided with the tool selects system data records with an owner flag of **CM** and with a **CM** prefix.

> **Note**
>
> If you choose not to follow the CM naming convention for your records and you want to use the CM packaging tool, your implementation must customize the extract file to define the appropriate selection criteria for the records to be included in the package. Refer to the Software Development Kit documentation for more information.

Also note that owner flag may be introduced to an existing table in a new release. When this happens, the CM packaging tool is also updated to include these new system data tables. Your implementation will have existing records in those tables that probably do not follow any naming convention. After an upgrade to such a release, if you want to include this data in the CM packaging tool, you must customize the extract file for the tables in question.

# Accessibility Considerations

This section provides information about topics related to accessibility that should be considered when extending the system via configuration tools.

### Color Contrast

The HTML color `red` (RGB code #FF0000) when used for text on a white background does not comply with the accessibility color contrast guidelines. When it is desired to use red for text for emphasis in a zone, map or in a text string, the product recommends the shade of red with the RGB code #E0292F.

Note that in addition, the product provides a special CSS class **textColorRedOnWhite** which may be used when HTML references are used for applying the red color. For example, UI map, zone help text and scripting.

### Identifying Rows in Grids

When navigating between rows in a grid on the user interface, screen readers will read out the row number and the element name (or names) with an attribute of 'scope=row'. By default, the system assigns this attribute to the non-suppressed primary key element (or elements) or to the first element in the grid if no primary key is defined. This may not be the information that best identifies the row. For example, a sequence number may be less meaningful than the heading of the next element in the list, such as 'Parameter Name'. The product provides a UI


Copyright © 2026, Oracle and/or its affiliates.

hint to allow you to specify one or more elements that should be used as the row header for accessibility.

Refer to UI Hint syntax for more details.

# Referencing URIs

There are some configuration objects that require a reference to a URI, including file path URIs. The following sections highlight some functionality supported by the product with respect to defining / accessing URIs.

> **Note**
>
> In order for the functionality described below to occur, specific APIs must be used by the underlying code related to the fields that capture or process the file path or URL information. If you find that there is a URI-related field that does not provide the functionality described here, please contact customer support.

> **Note**
>
> For schema elements in a business object that reference a Field with the **URI** data type or define the element with the **URI** data type configured in the schema will automatically use the appropriate API that validates the value that may reference a substitution variable or must be checked against the allowlist if applicable.

- Validation Against an Allowlist
- URI Substitution
- External File Storage

## Validation Against an Allowlist

Based on a property setting, your implementation may be configured to define a allowlist of URIs. If this setting is enabled, the system will issue an error if the URI is not defined in the allowlist. Consult your system administrator to verify if this setting is enabled or not for your implementation.

> **NOT_SUPPORTED**
>
> Refer to URI Substitution for information about how defining substitution tokens for URIs using the properties file technique automatically adds the defined URI to the allowlist.

## URI Substitution

The system supports the ability to define substitution variables (sometimes referred to as tokens) for both URL values and file path values that reference native file storage locations. For URLs, the system supports defining variables in a Substitution Variable properties file. For native file storage paths, there are two options: the Substitution Variable properties file or via a File Storage configuration extendable lookup. More details on both these options are found below.


Copyright © 2026, Oracle and/or its affiliates.

- Substitution variables properties file. A substitution variable for all or part of the URI definition may be configured in a properties file. This allows the system administrators to define the proper URI locations in a properties file whereas the configuration users only need to know the variable name. For example, when defining a location for an extract file in an extract batch job, instead of typing a file path of `h:\oracle\serverName\1.0.0.0\batch\extract\`, the batch user can enter `@FILE_EXTRACT@`, assuming there is an entry in the substitution variables file with a name of `FILE_EXTRACT`, and a value of `h:\oracle\serverName\1.0.0.0\batch\extract\`. Another example is that the batch user could enter `@BATCH_FILES@\extract\`, assuming that the URI variable for `BATCH_FILES` is defined as `h:\oracle\serverName\1.0.0.0\batch\`.

  > **Note**
  >
  > The product automatically populates the value of SPLOUTPUT in the properties file so that this may be used in URI configuration. In addition, the product may supply some pre-defined variable names for other common references. As part of this, the 'advanced' menu in the system installation steps may prompt for installers to define the values of these pre-defined variables, if desired. Installations may opt to define additional substitution variables for various URI references. Refer to the *System Administration Guide* for more information.

- File Storage extendable lookup. Specifically for file paths that reference the native file system, the system also supports the ability to define a path using the **File Storage Configuration** extendable lookup. The following points highlight the steps to take for this option.
  - Navigate to Extendable Lookup and search for the **File Storage Configuration** lookup.
  - Click **Add** to create a new entry. Define a lookup value name. This will be used when configuring a file path that uses this value. Choose the File Adapter value of **Native File Storage**. Enter the desired file path value. The values defined here may in turn refer to values defined in the Substitution Variables properties file. For example, the file path can reference @SPLOUTPUT@.
  - To reference this value in system configuration, use the syntax `file-storage://XXXX`, where XXXX is the extendable lookup value. Using the same example from above, if you define an extendable lookup value of CM-FileExtract with a file path of `h:\oracle\serverName\1.0.0.0\batch\extract\`, then when configuring the extract file path for an extract batch job, enter `file-storage://CM-FileExtract`. Another example is that if the extendable lookup value's file path is configured as `h:\oracle\serverName\1.0.0.0\batch\`, then the user configuring the file path on the batch extract can enter `file-storage://CM-FileExtract/extract`

When should you define substitution variables in the properties file and when you define them in the extendable lookup? The following points highlight differences between the options that may help this decision.

- Substitution variables for URLs are only supported via the properties file.
- The properties file can typically only be modified by a system administrator. If there are values that are set at installation time and don't change, then defining the values in the properties file may be beneficial.
- When defining additional environments, such as test environments or production, the values in the extendable lookup may be copied using CMA. Ideally the values are defined such that they are the same between various regions.


Copyright © 2026, Oracle and/or its affiliates.

- When attaching files while Sending Email, if the files are located in native file storage, the file path must be declared in an entry in the File Storage Configuration extendable lookup.

## External File Storage

The system supports using Oracle Cloud Object Storage for managing files in cloud implementations and can be configured to read files from or write files to this external location. Refer to Oracle Cloud Object Storage documentation for more information about obtaining an account and defining appropriate file locations or "buckets".

Once your cloud storage information is defined, the following points highlight the configuration steps required in the application.

- Define a signature key ring. When the system tries to communicate with cloud object storage, it must provide a signature key so that cloud storage can confirm that the request is from a trusted source.
  - Navigate to Key Ring in add mode and select the **RSA Signature Key Pair** business object.
  - Define a key ring code, which will be used in the File Storage configuration (below) along with a description.
  - Once the key ring is added, click **Generate Key** to generate a private / public key pair.
  - Click View Public Key to launch a pop-up that displays the public key, allowing the user to copy the key.
- At this point, the user should navigate to cloud object storage and register the public key. Refer to Cloud Services Administration Guide for more information on this step. Once this is done, the key ring is now ready to be configured in file storage configuration.
- Next, confirm that the region where your object storage is defined is configured in the system and if not, define the value.
  - Navigate to Extendable Lookup and search for the **OCI Region** lookup. Review the list of regions already provided.
  - If your region is not listed, click the Add button. Be sure that the Region Key and the Region Identifier match the values defined in the OCI region documentation. The region key is a three digit identifier. For example, the one for Sydney region in Australia is SYD. The region identifier follows a pattern of a two digit realm id followed by the city name, followed by a single digit, all separated by hyphens. For example, the one for the Sydney region is "ap-sydney-1".
- Use a File Storage extendable lookup to define the location and connection information so that you may reference this location in system configuration. The following points highlight the steps to take for this option.
  - Navigate to Extendable Lookup and search for the **File Storage Configuration** lookup.
  - Click **Add** to create a new entry. Define a lookup value name. Note that the lookup value should not have a slash or backslash in its name. This will be used when configuring a file path that uses this value. Choose the File Adapter value of **Oracle Cloud Object Storage**. Provide the following information that identifies the cloud storage options: User, Tenancy, Compartment, Namespace and Region. For the Key Ring, choose the value defined above.
- To reference this value in system configuration, use the syntax `file-storage://XXXX/...`, where XXXX is the extendable lookup value and any additional path information that is appropriate. For example, if you define an extendable lookup value of


Copyright © 2026, Oracle and/or its affiliates.

`CM-CloudStorage` and you have a bucket defined in Cloud Storage for processUpload, when configuring the file path for the upload batch job, enter `file-storage://CM-CloudStorage/processUpload`.

- The product supports referencing virtual folders for a bucket. Simply add the virtual folder name(s) after the bucket name. For example, if there is a bucket called 'financials' and a virtual folder for 'paymentUpload', the file path may have this value: `file-storage://CM-CloudStorage/financials/paymentUpload`. For extract processes, if the file path references a virtual folder that does not exist for the referenced bucket, the virtual folder is created as part of the extract process.
- You may optionally use a Bucket Name Prefix on the file storage configuration to support defining different Bucket Names for different environments. For example, imagine you want to have a separate "processUpload" bucket for a development region, a test region and a production region. In the Cloud Storage configuration you can define separate buckets for "DEV_processUpload", "TEST_processUpload" and "PROD_processUpload". If you needed to define the full bucket on each file path reference on various batch jobs, migrating the batch job configuration from one environment to another would require a manual step to change all the bucket references. Instead, the system supports defining a Bucket Name Prefix, which if populated will be prepended to bucket reference before connecting to Cloud Storage. This means that the prefix is defined once and all the references throughout the system can keep the common part of the Bucket name (such as "processUpload"). Then only the File Storage configuration needs to differ between the different regions.
- If your implementation integrates with an external reporting tool that can write the report output to specific delivery channels (buckets) check the Reporting Configuration switch. Define one or more Bucket Names for the buckets that the reporting tool may use for delivery channel options.

> **Note**
>
> Refer to the Key Ring documentation for information about practicing key rotation.

# Caching Overview

A great deal of information in the system changes infrequently. In order to avoid accessing the database every time this type of information is required by an end-user or a batch process, the system maintains a cache of static information on the web server and in the batch thread pool worker. These are referred to as the "application caches". Some examples of application caches include

- System messages
- Field label and other field information
- Security Information

The framework product provides many specific caches for commonly used (and infrequently changed) data. In addition, specific edge applications may introduce additional caches as appropriate.

Information may also be cached on each user's browser.

The following topics highlight information about refreshing the various caches.


Copyright © 2026, Oracle and/or its affiliates.

### Server Cache

The server cache refers to data that is cached on the web server. An important use of this cache is for users' online access to the application. The caches aid in better performance while navigating throughout the system, allowing for data to be accessed from the cache rather than by always accessing the database. Besides user access to the web server cache, other functionality deployed to the web server uses caches in a similar way. For example, web services are deployed to the web server and access their own version of the cache.

The contents of the cache are cleared whenever the web server is restarted. This means that fresh values are retrieved from the database once users and web services start using the application again.

The product also supplies a flush command that one can issue in the browser's URL to immediately clear the contents of the cache. The command `flushAll.jsp` flushes every cache (with an exception noted below).

For example, assume the following:

- the web server and port on which you work is called `OU-Production:7500`
- you add a new record to a control table and you want it to be available on the appropriate transactions immediately

You would issue the following command in your browser's address bar: `http://OU-Production:7500/flushAll.jsp`. Notice that the command replaces the typical cis.jsp that appears after the port number.

If your system has been configured correctly, the **flushAll** command will submit a request to do a "global" flush of caches (including the web services cache and the thread pool worker cache). This functionality uses a JMS Topic to publish the flush request. Refer to the *Server Administration Guide* for details on how to configure the JMS topic.

> **Important**
>
> For a cloud / production environment, the flushAll command will not flush the script cache. This is because flushing the script cache requires an entire recompilation of all the scripts. You don't want to inadvertently cause script recompilation in production when a script hasn't been changed. When a script has actually been updated in production, the script cache can be explicitly flushed by adding "?cacheNames=ScriptCache" to the URL. For example: `http://OU-Production:7500/flushAll.jsp?cacheNames=ScriptCache`

Also note that the system provides a business service `F1-FlushAllService`, which is equivalent to the flushAll command (except that it does not restart your browser session). This may be useful for any processes that are built to orchestrate updates to configuration data where you want to ensure that the cache is immediately flushed. This business service is also exposed as a REST web service.

> **Note**
>
> This business service does not flush the script cache in a cloud production environment. You need to use the browser command noted above to flush the script cache in this case.


Copyright © 2026, Oracle and/or its affiliates.

### Batch Cache

When submitting a batch job, the batch component uses a Hibernate data cache to cache administrative data that doesn't change very often. The tables whose records are included in this cache are configured using the Caching Regime value of **Cached for Batch**. This is often referred to as the "L2 cache". Refer to Tables for more information. When starting a thread pool worker, data in tables marked as cached is loaded and cached for as long as that thread pool is running.

In addition batch jobs may also access application caches when applicable. When starting a thread pool worker, application data that is cached is loaded and cached for as long as that thread pool is running.

If there are any batch jobs that update any tables that are marked as **Cached for Batch**, you must run that batch job using a special thread pool that is configured with the caching disabled. Contact your system administrator to confirm the thread pool that is configured this way.

If there is a change in cached data that should be available for the next batch job, the following points highlight how the cache can be refreshed:

- By default the system is configured to automatically refresh the Hibernate cache every 60 seconds. However, an implementation may override the configuration to either change the number of seconds between intervals or to disable the automatic caching altogether. Application caches used by the batch jobs are not impacted by this refresh.
- Restart the thread pool workers.
- Run the `F1-FLUSH` (Flush all Caches) background process. This background process will flush the application data cached for all thread pool workers for all thread pools.
- If your the region has configured the thread pool workers to "listen" to requests for global flush as described in the Server Cache section, the thread pool worker caches are also refreshed when a **flushAll** command is issued.

> **Note**
>
> The Flush all Caches background process does not flush the script cache in a cloud production environment. You need to use the browser command noted above to flush the script cache in this case.

### Client Cache

In addition to the web server's cache, information is also cached on each user's browser. After clearing the cache that's maintained on the web server, you must also clear the cache that's maintained on your client's browser. To do this, follow the following steps (note that depending on the browser you use, the steps may vary slightly):

- Select **Tools** on your browser's menu bar
- Select **Internet Options...** on the menu that appears.
- Click the **Delete Files** button on the pop-up that appears.
- Turn on **Delete all offline content** on the subsequent pop-up that appears and then click **OK**.
- And then enter the standard URL to re-invoke the system.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Automatic refresh of the browser's cache.** Each user's cache is automatically refreshed based on the **maxAge** parameter defined in the web.xml document on your web server. We recommend that you set this parameter to `1` second on development / test environments and `28800` seconds (8 hours) on production environments. Please speak to system support if you need to change this value.

# Expression Parser

The product provides support for defining expressions that may be of a mathematical or logical/boolean nature. The expression may include variables and functions.

The data explorer column parameter is an example of where this may be used. That parameter supports the definition of a formula. Edge applications may include support for a formula or expression using this parser as well. For example, several application include a type of 'rule' object (calculation rule, form rule or usage rule) that is used for validation or calculation that may support applying a formula.

The following tables highlight what is supported in the expressions that use this parser.

| Category | Supported in Expression | Description |
|---|---|---|
| Data types | Number | |
| | String | |
| | Boolean | |
| | List | |
| Literals | Numbers | |
| | Strings surrounded with either single quote or double quote.<br><br>**Note**<br>'Escaping' special characters is not currently supported. | |
| | Boolean values: **true** and **false**. | |
| Operations | + | Plus |
| | — | Minus |
| | / | Division |
| | * | Multiplication |
| | ^ or ** | Power |
| | % | Modulus |
| Logical operations | = | Equal |
| | > | Greater than |
| | >= | Greater than or equal to |
| | < | Less than |
| | <= | Less than or equal to |


Copyright © 2026, Oracle and/or its affiliates.

| Category | Supported in Expression | Description |
|---|---|---|
| | != or <> | Not equal to |

This table identifies the functions that are supported. Note that several of the functions are applicable to a list of values. Note that although the functions are listed in lower case, the column parameter syntax in data explorer indicates referencing the functions as all capital letters. The system converts the data explorer column formula to lowercase before being evaluated.

| Function | Parameter | Results | Comments |
|---|---|---|---|
| `size( )` | List element | Number of elements in the list. | |
| `isEmpty( )` | List element | Returns true if the list is empty. | |
| `sum( )` | List element of type 'number' | Returns the sum of the numbers in the list. | |
| `avg( )` | List element of type 'number' | Returns the average of the numbers in the list. | |
| | One or more numbers separated by commas | Returns the average of the number arguments. | |
| `max( )` | List element | Returns the largest value in the list. | |
| | One or more comparable elements. | Returns the largest value of the number arguments. | |
| `min( )` | List element | Returns the smallest value in the list. | |
| | One or more comparable elements. | Returns the smallest value of the number arguments. | |
| `abs( )` | Number | Returns the absolute value. | |
| `ceiling( )` | Number | Rounds the number to the ceiling. | |
| `exp10( )` | Number | Raises 10 to the number power. | |
| `acos( )` | Number | Returns the arc cosine of the number in radians. | The result will lose precision, as it uses the system's double float based functions. |
| `asin( )` | Number | Returns the arc sine of the number radians. | The result will lose precision, as it uses the system's double float based functions. |
| `atan( )` | Number | Returns the arc tangent of the number radians. | The result will lose precision, as it uses the system's double float based functions. |
| `cos( )` | Radian | Returns the cosine of the radian angle input. | The result will lose precision, as it uses the system's double float based functions. |
| `exp( )` | Number | Raises e to the number power. | The result will lose precision, as it uses the system's double float based functions. |
| `log10( )` | Number | Takes the log, base 10, of the number. | The result will lose precision, as it uses the system's double float based functions. |


Copyright © 2026, Oracle and/or its affiliates.

| Function | Parameter | Results | Comments |
|---|---|---|---|
| `log( )` | Number | Takes the natural log (base e) of the number. | The result will lose precision, as it uses the system's double float based functions. |
| `sin( )` | Radian | Returns the sine of the radian angle input. | The result will lose precision, as it uses the system's double float based functions. |
| `sqrt( )` | Number | Returns the square root of the number. | The result will lose precision, as it uses the system's double float based functions. |
| `tan( )` | Radian | Returns the tangent of the radian angle input. | The result will lose precision, as it uses the system's double float based functions. |
| `floor( )` | Number | Rounds the number to the floor. | |
| `round( )` | Number | Assumes a scale of 0. The default rounding mode of “round half up” is applied. | |
| | Number, Scale | The default rounding mode of “round half up” is applied. | |
| | Number, Scale, Mode | The mode must be set to one of the following:<br>• `“ROUND_CEILING”`<br>• `“ROUND_DOWN”`<br>• `“ROUND_FLOOR”`<br>• `“ROUND_HALF_DOWN”`<br>• `“ROUND_HALF_UP”`<br>• `“ROUND_HALF_EVEN”`<br>• `“ROUND_UP”`<br>• `“ROUND_UNNECESSARY”` | |
| `negate( )` | Number | Returns the negative value of the number. | Only available in data explorer. |

The following are special functions supported in the application for a list of values. In each case, the syntax is *function* [ *indexVariable* in *listName* | *expression using indexVariable* ], where the *indexVariable* is chosen by the formula writer to represent each entry in the list and the expression used to evaluate each entry must reference that variable.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The syntax supported for a given use of the formula in a functional area is driven by that particular functional area. For example, in Oracle Public Sector Revenue Management, a formula is supported in the “conditional element validation” form rule. In that form rule all variables including lists are declared in the form rule using letters and the formulas in turn use these letters. In that scenario, the functions below would reference the declared variable letter as the “listName”. Other specific functional area that use this expression parser may support different syntax for referencing elements or lists.

| Function | Description | Examples |
|---|---|---|
| `any [ ]` | This function returns the value **true** if any of the entries in list satisfies the expression. | The following returns **true** if any entry in the Balance list is greater than 0.<br>`any [ i in list/Balance \| i > 0 ]` |
| `all [ ]` | This function returns the value **true** if all of the entries in the list satisfy the expression. | The following returns **true** if all phone numbers are populated.<br>`all [ i in list/phoneNumber \| i != ' ' ]` |
| `collect [ ]` | This function returns a new list of elements from the referenced list where the value of each entry of the new list is the result of the expression applied to each original value. | The following returns a new list with the tax rate applied to each amount.<br>`collect [ i in list/amount \| i * taxRate ]` |
| `select [ ]` | This function returns a list of all the values of the original list that satisfy the Boolean expression. | The following returns a new list with only the amounts that are negative numbers.<br>`select [ i in list/amount \| i < 0 ]` |
| `reject [ ]` | This function returns a list of all the values of the original list that do not satisfy the Boolean expression. | The following returns a new list with only the amounts that are not negative numbers.<br>`reject [ i in list/amount \| i < 0 ]` |

# Debug Mode

Your implementation team can execute the system using a special mode when they are configuring the application. To enable this mode, enter `?debug=true` at the end of the URL that you use to access the application. For example, if the standard URL was `http://CD-Production:7500/cis.jsp`, you'd enter `http://CD-Production:7500/cis.jsp?debug=true` to enable configuration mode.

In addition, if you have security access to the application service `F1OUDBUG`, you can enable this mode without changing the URL via the Help Menu - Enable Debug option. If debug mode


Copyright © 2026, Oracle and/or its affiliates.

is enabled, the help menu entry shows a Disable Debug option, allowing you to turn it off without changing the URL.

When in this mode certain debugging oriented tools become available via an icon visible. Clicking the icon displays a slide-out panel with options defined below.

The following are related to the advanced debugger:

- **Start Debug** starts a logging session. During this session the processing of various configuration tools objects are logged. For example, the log will show the data areas that are passed in at each step and the data areas returned after the step is processed.
- **Stop Debug** stops the logging session.
- **Show Trace** opens a window that contains the logging session. All of the steps are initially collapsed.
- **Clear Trace** clears your log file.

> **Note**
>
> **The buttons are secured.** An application service `F1CONFIGLOG` has been provided for this functionality to allow implementations to restrict user access to these buttons. An additional application service `F1DEBUG` is checked when the environment is not a 'development' environment and is only applicable on premises. This allows implementations to provide an additional layer of security for their production users. (For cloud customers, the advanced debugger is only available in a development environment).

The following are available to view log entries captured while using the system.

- **Show User Log** allows you to view your own log entries. The number of "tail" entries to view may be specified in the adjacent **Log Entries** field before clicking the button. Limiting the number of entries to view allows the user to quickly and easily see only the latest log entries without having to manually scroll to the end of the log.
- **Show Log** allows for advanced settings. It provides for a date / time range to allow a user to target the output for a specific period. In addition, administrative users may request to view logs of other users. The following are examples of when this capability is needed:
  - Batch and integration logs are captured by special user codes created for this type of processing. These users typically do not represent a human and therefore wouldn't access logs using "show user log". Instead someone else would be accessing these logs.
  - Customer admins would like to retrieve logs for an end user possibly for support reasons. An end user might not have technical skills to gather or assess the logs nor would they be given the privilege to access the logs for security reasons
- Checking the **Global Debug** indication starts various tracing options that are added to the user log.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Show User Log button is secured.** An application service `F1USERLOG` has been provided for this functionality to allow implementations to restrict user access to this button. Only users with access to the **Administrative** access mode of the application service are allowed to view logs of other users. Such restrictions may be called for in production environments.

> **Note**
>
> If your environment configures redaction rules, personal identifiable information may be redacted in the log output. Refer to Data Redaction for Export and Logging for more information.

For Cloud implementations only, the following buttons are available in this mode:

- **Show Access Logs** - Clicking this button launches a window allowing you to define a date / time range and the number of "tail" entries to view. Click 'Save' to view the access logs that satisfy the criteria.

  > **Note**
  >
  > **Show Access Log button is secured.** This button is secured using the application service `F1ACCLOG` and **Administrative** access mode.

- **Restart TPWs** - This button forces batch containers to terminate, interrupting all running batches. New containers will spin up and the interrupted batches resume. Refer to the Batch Best Practices technical reference paper for more information about this button and when to use it.
- **Restart CM Jar Servers** - This button is provided to aid with support in the cloud for custom Java code in certain use cases. Refer to the Cloud Services Implementation Guide for more information.

Other parts of the system may show additional configuration oriented icons when in this mode. For example, explorer zones may provide additional tools to assist in debugging zone configuration. These icons are described in the context of where they appear.

Also, in debug mode drop down lists in data explorer and UI map zones will contain the code for each item in addition to the item's display string.

# Prepare Issue Details

When a user encounters an error with the product and submits a service request, information about the environment, the release and the configuration of the user is often very helpful in understanding the cause of the problem. The system provides a script to capture the environment details and issue description for reporting purposes.

To access the script use **Menu**, and then **Tools**, and then **Prepare Issue Details**. The option is also available in the help menu in the toolbar.

A pop-up window is provided to display the details that are being captured, including a generated Issue ID to uniquely identify the report. The pop-up allows the user to add additional information describing the specific issue. When the user presses Save, the details are added


Copyright © 2026, Oracle and/or its affiliates.

to the system log with the issue ID, which can be searched when investigating the problem. A message will be displayed to either confirm the details were successfully logged or indicate that an error occurred.

> **Note**
>
> **The script and menu item are secured.** An application service `F1-PREPISSUEDTLS` has been provided to allow implementations to restrict user access to this functionality.

## System Override Date

The system provides a way to override the system date used for online operations. This feature is available if the server administrator has enabled it in the environment properties. For instructions on configuring environment properties see the *Server Administration Guide*. The system date override feature is not recommended for production environments.

Under the **General System Configuration** Feature Configuration, the **System Override Date Option Type** holds the date the application will use as the global system date instead of retrieving the same from the database. This feature can be especially useful in running tests that require the system date to be progressed over a period of time.

The system override date feature is also available at the user level. This is useful when a user wants override the system date to run tests without affecting the system date for other users in the environment. In order to override the system date for the user, open the User — Characteristics page, add the **System Override Date** characteristic type with a characteristic value set to the desired date in the YYYY-MM-DD format.

If system override dates are defined at both the feature configuration level and the user level, the date set at the user level will take precedence.

## Additional Search Options

The product supports fuzzy searching in explorer zone types using the Oracle Text CONTAINS operator.

Refer to the DBA guide for details on setting up the database to support fuzzy searching. Note that there are some implementations where fuzzy searching will not be possible. For example, it's only available for implementations using the Oracle database. Additionally, not all languages are supported. Refer to the Oracle Database documentation for more information about fuzzy searching.

For information about the particular syntax to use in the explorer zones, refer to SQL Statement in the zone parameter details section.

# Extensions Dashboard

This portal provides a high level summary of configuration entities that have been customized and extended to meet your business requirements. Included are customer owned entities and extended product owned entities. The portal is intended to be used by your implementation team to get an appreciation of areas that have been customized and may need to be reviewed or retested for various reasons.

Use **Admin**, and then **Implementation Tools**, and then **Extensions Dashboard** to navigate to the portal.


Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page

- **Extensions Summary.** This zone provides a summary of customized entities by maintenance object. You may select a specific maintenance objects to review entities of this type.
- **Extension Entities.** This zone lists customized entities of the selected maintenance object. It is only visible when a maintenance object is selected on the Extensions Summary zone.
- **Extension Details.** This zone provides additional information as to which specific tables contain the custom owned content for the selected entity. It is only visible when a specific customized entity is selected.
- **Revision Control.** This zone lists revisions made to the selected entity. It is only visible when revision information exists for the selected entity.
- **Configuration Migration.** This zone lists CMA migration objects that applied changes to the selected entity. It is only visible when import information exists for the selected entity.

The following zone may appear as part of the portal's **Review** tab page

- **Duplicate Configurations.** This zone highlights duplicate configuration issues that may rarely arise after an upgrade. Customers should use this information to adjust thir configuration accordingly.

# To Do Lists

Certain events that occur within the system will trigger messages describing work that requires attention. For example, if a bill segment has an error, the system generates a To Do message to alert the person responsible for correcting such errors.

Each type of message represents a To Do list. For example, there are To Do lists for bill segment errors, payment errors, customer contact reminder, etc.

We refer to each message as a **To Do Entry**. Each To Do entry is assigned a specific **To Do Role**. The role defines the users who may work on the entry. A To Do entry has a **To Do log** that maintains record of the progress on the To Do entry. For example, the To Do log indicates when the To Do entry was created, when it was assigned to a user and to whom it was assigned, and when and by whom it was completed.

**NOT_SUPPORTED**

Refer to To Do Processing for a description of end-user queries and tools assisting in reviewing, assigning and processing To Do entries.

- The Big Picture of To Do Lists
- Setting Up To Do Options

## The Big Picture of To Do Lists

The topics below provide more information about To Do configuration.

- To Do Entries Reference A To Do Type
- To Do Entries Reference A Role


Copyright © 2026, Oracle and/or its affiliates.

- The Lifecycle Of A To Do Entry
- How Are To Do Entries Created?
- Refining a To Do Entry's Details at Creation Time
- The Priority Of A To Do Entry
- What Happens When You Work A To Do Entry
- Monitoring A To Do Entry
- To Do Entries Have Logs
- Implementing Additional To Do Entry Business Rules
- To Do Entries May Be Routed Out Of The System
- To Do Information May Be Formatted By An Algorithm
- Periodically Purging To Do Entries

## To Do Entries Reference A To Do Type

Every To Do entry references a To Do type. The To Do type controls the following functions:

- The page into which a user is taken when they drill down to the related object on an entry.
- The message associated with the To Do that appears on various pages. Note this message can be overridden for specific To Do messages by specifying a different message number in the process that creates the specific To Do entry. For example, the process that creates To Do entries associated with bill segments that are in error displays the error message rather than a generic "bill segment is in error" message.
- The To Do entry sort keys. Note that the processes that create To Do entries are responsible for populating the sort key values.
- Whether (and how) the To Do entry is downloaded to an external system (e.g., an email system).
- The roles to which an entry may be reassigned.
- The default priority of the To Do entry. Note that this value may be overridden by a **Calculate Priority** algorithm.
- An indication of whether a To Do of that type may be created manually by a user.
- The algorithms used to perform specific business rules for To Do entries of this type.
- The characteristics applicable to the To Do.

## To Do Entries Reference A Role

Every To Do entry references a role. The role defines the users who may be assigned to **Open** entries.

The permissible roles that may be assigned to a To Do entry are defined on the entry's To Do type. After an entry is created, its role may be changed to any role defined as valid for the entry's To Do type.

The base package determines the initial To Do Role as follows:

- The system checks if an entry's message category / number is rerouted to a specific role. If so, it defaults this role. Refer to Adjusting the To Do Role for more information.


Copyright © 2026, Oracle and/or its affiliates.

- There may be some other condition related to the object linked to your To Do entry that warrants an override role. A To Do pre-creation plug-in would be used to check the conditional logic and override the role, if applicable.
- If a Role wasn't determined in one of the previous steps and a Role is provided by the initiating process (algorithm or background process), the entry is created with that Role.
- If the entry does not have a role after the above takes place, the entry's To Do type's default role is assigned to the entry.

> **Note**
>
> At installation time, the system provides a default role assigned to the system To Do types when first installed called `F1_DFLT`. This is done to allow testing of the system prior to implementing of appropriate To Do roles for your organization. The recommendation is to configure all the To Do Types with appropriate business oriented To Do roles once they are defined.

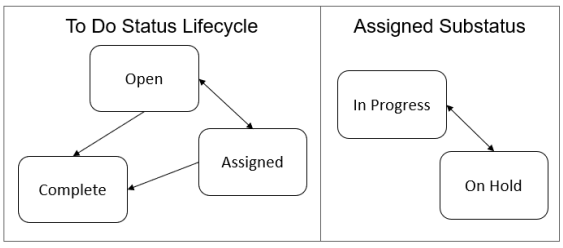
**Supervisors for a Role**

Most organizations have the notion of a supervisor who is responsible for all entries assigned to a given role. A user that is considered a supervisor should be a valid user in a role. In addition, application security is used to provide supervisors access to additional capabilities, namely the ability to review To Do entries assigned to other users and the ability to assign To Do entries to other users.

The pages listed in To Do Supervisor Functions are meant for supervisors only. So security for these pages should be restricted to those users that are considered supervisors. The To Do Management portal and the To Do Search page are available to all users. There is a special access mode for Supervisors to allow those users the ability to assign To Do entries to other users.

# The Lifecycle Of A To Do Entry

The following state transition diagram will be useful in understanding the lifecycle of a To Do entry.




A To Do entry is typically created in the **Open** state. Entries of this type can be viewed by all users belonging to the entry's role. Refer to How Are To Do Entries Created? for information about how entries are created.

An **Open** entry becomes **Assigned** when it is assigned to a specific user or when a user proactively assumes responsibility for the entry. While an entry is **Assigned**, it is visible on the To Do Summary page only by the user who is assigned to it.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **To Do entries may be created in the Assigned** state. Some To Do background processes may create To Do entries in the **Assigned** state. When a user adds a To Do entry online and assigns the To Do to a user (as opposed to a role), the To Do entry is also created in the **Assigned** state.

An **Assigned** entry may be forwarded to a different user or role. If the entry is forwarded to a role, it becomes **Open** again. This may also be referred to as "reopening" or "unassigning" a To Do entry.

While a To Do Entry is in the **Assigned** state, if your implementation has configured it, you may further indicate the substatus of **In Progress** or **On Hold**.

- When you are ready to start work on an entry you can put it in the **In Progress** state. Refer to Working On A To Do for more information on how to do this.
- If you are not able to complete the To Do entry, you can put it **On Hold**, or stop working on it to put it back to the **Assigned** state (basically resetting the substatus). In either case, the To Do entry is still assigned to you.
- You may only have one To Do entry in the **In Progress** state. If you already have an entry in this state and you perform any action that puts a new To Do entry in **In Progress**, the other To Do entry will automatically be put into the **On Hold** state.

> **Important**
>
> To configure the system to allow users to use the substatus capability, you simply need to give users security for the Start Progress and Hold Progress and Stop Work access modes on the To Do Entry application service (**CILQTDEP**). The recommendation is that an implementation either grants all To Do users access to these access modes or grants no users access to these access modes.

> **Note**
>
> The substatus values are captured in a separate field (`TD_SUB_STATUS_FLG`). They are only populated if the main status field (`ENTRY_STATUS_FLG`) is in the Assigned status (`W`). But most pages in the system treat these substatus values as another status value. You typically won't see both values Assigned and In Progress or Assigned and On Hold. You'll just see In Progress and On Hold.

When an entry becomes **Complete**, it is no longer visible in various To Do queries (but it remains on the database for audit purposes). There are several ways an entry can become **Complete**:

- A user can manually indicate it is **Complete**.
- There are some background processes provided that create To Do entries based on the condition of some object. When subsequently running the same background process, the system automatically marks the To Do entry **Complete** when the object is no longer in the given condition. For example, an entry that's created when an account doesn't have a bill cycle is completed when the account has a bill cycle.


Copyright © 2026, Oracle and/or its affiliates.

> **Important**
>
> The automatic completion of To Do entries occurs when the background process responsible for creating entries of a given type is executed. Therefore, depending on how often that process is run, there may be a delay where a To Do entry remains in a non-complete state even if the reported condition no longer exists.

- Algorithms are provided for lifecycle based objects that can create a To Do entry when entering or exiting a status. And corresponding algorithms may be used to complete the To Do entry when entering or exiting a status.

> **Note**
>
> The base product provided algorithm to complete a To Do Entry when entering or exiting a status includes the ability to configure that To Do entries of a certain type should not be automatically completed. These algorithms rely on a characteristic type of F1-EXLTD configured on the To Do Types that should not be auto-completed.

- The To Do monitor process may be used to evaluate whether or not the To Do entry can be closed automatically. Refer to Monitoring a To Do Entry for more information.

## How Are To Do Entries Created?

A To Do Entry may be created in the following ways:

- A background process can create To Do Entries.
- An algorithm can create entries of a given type. Because the use of algorithms is entirely dependent on how you configure the control tables, the number of types of such entries is indeterminate.
- A user can create entries of To Do types that have a **Manual** usage. Refer to To Do Entries Created Manually for information about setting up manual To Do types.

For any base product process that includes logic to create a To Do entry, the system supplies a sample To Do type that may be used. Although the To Do types provided by the product are system data, the following information related to each To Do type may be customized for an implementation and is not overwritten during an upgrade:

- The creation process. If the To Do is created by a background process where the background process is referenced on a To Do type. Refer to To Do Entries Created By Background Processes for more information.
- The routing process. Refer to To Do Entries May Be Routed Out of the System for more information.
- The priority. Refer to To Do Type for more information.
- The roles that may be associated with the To Do type. Refer to To Do Entries Reference a Role for more information.
- The message override information. Refer to Suppress the Creation of the To Do Entry, Adjusting the To Do Role and What Happens When You Work A To Do Entry for more information.
- Suppress the Creation of the To Do Entry
- To Do Entries Created By Background Processes


Copyright © 2026, Oracle and/or its affiliates.

- To Do Entries Created By Algorithms
- To Do Entries Created Manually

## Suppress the Creation of the To Do Entry

The system provides the ability to indicate that a To Do Entry should not be created based on conditions. When the system detects that no To Do should be created, there is no error. The system simply returns to the caller with an indication that no To Do was created. This functionality only applies to automatic To Do Entries created by an algorithm or background process. It does not apply to manually created To Do Entries.

### Suppress Based on To Do's Message

There are some types of To Do Entries that are generated with a message that is specific to a condition that was found for a record. For example, business errors found when trying to update a record through a system process will generate a To Do entry using the error message as the To Do Entry's message. There may be some use cases where the error reported is not something that a user needs to address and it may be beneficial to your organization to suppress the To Do Entry. Note that this is only appropriate if the message generated is for informational or warning purposes and doesn't require action or if you know that the error will be resolved by a subsequent step.

The configuration to suppress a To Do entry based on its message is on the To Do type's message overrides page.

> **Note**
>
> **Message Category / Message Number.** Every error in the system has a unique message category / number combination. Refer to The Big Picture of System Messages for more information about message categories and numbers.

We do not supply documentation of every possible message that can be handled by a given To Do type. The best way to build each To Do type's reroute list is during the pre-production period when testing the system. During this period, compile a list of the messages that warrant special behavior described here.

Note that if the To Do type is configured to reference a specific message category / number, it means all To Do entries have the same message. If your organization does not want To Do entries created for this To Do type, then rather than using this suppression technique, you should identify the background process or algorithm that generates the To Do entries of this type and change the configuration there such that it does not try to create the To Do in the first place.

### Suppress Based on a Condition

There may be use cases where a To Do entry should be suppressed based on some conditional logic. For example, consider the billing process in Oracle Utilities Customer Care and Billing. There is typically a bill cycle schedule that allows for attempting to bill all the accounts for the bill cycle over a multi-day window. There are some errors that occur on day 1 of the window that may be resolved by day 2 or 3. An implementation may choose to suppress To Do entries for this condition if this is day 1 or day 2 of the bill cycle. If the error still occurs on day 3, then the To Do entry should be generated. This type of condition needs to be determined using an algorithm. The To Do pre-creation plug-ins support returning an indication to suppress the To Do entry. Refer to To Do Pre-creation Plug-ins for more information.


Copyright © 2026, Oracle and/or its affiliates.

# To Do Entries Created By Background Processes

There are different types of To Do entries created by background processes:

- To Do entries created by dedicated To Do background processes
- To Do entries created for object-specific errors detected in certain background processes
- To Do entries created based on a specific condition
- Dedicated To Do Background Processes
- To Dos Created for Object-Specific Error Conditions
- To Dos Created by Background Processes for Specific Conditions

## Dedicated To Do Background Processes

There are To Do entries that are created by system background processes whose main purpose is to create To Do entries based on a given condition. For these background processes, the To Do Type indicates the creation background process.

Note that the Plug-in Driven batch program cannot be used for this purpose. This program assumes that any To Do Type that references it as a creation process is using the link for the Object-Specific Error Condition use case. If you want to use the plug-in driven batch process to create To Do entries based on a given condition, then you need to follow the pattern described in To Dos Created by Background Processes for Specific Conditions.

> **Note**
>
> **If you don't schedule the background process, the entries will not be created!** The To Do entries of this type will only be created if you have scheduled the associated background process. Therefore, if you want the system to produce a given entry, schedule the background process.

## To Dos Created for Object-Specific Error Conditions

A system background process may create a To Do entry when an error is detected during object-specific processing. This is applicable for processes that do not have built in error handling, for example where there is an explicit "error" state or where the record has an explicit "exception" record.

For these background processes, the To Do Type must reference the creation background process.

To have the system create To Do entries for some or all of the errors generated by one of these processes, you must do the following:

- If you want the system to generate To Do entries for errors detected by one of the background processes below, go to the appropriate To Do type and populate the creation background process.
- If you want the system to generate To Do entries for some errors for the process, but not for all errors, populate the creation background process and then proceed to the message overrides tab to suppress certain messages. Do this by indicating the message category and message number you want to suppress. Any error that is suppressed is written to the batch run .


Copyright © 2026, Oracle and/or its affiliates.

The functionality will only create a new To Do entry if there is not already an existing (non-complete) To Do for the same To Do type, drill key and message category / message number. It will also check for an existing To Do for a successfully processed record and complete that To Do.

If you do not populate the creation background process, the errors are written to the batch run .

> **Note**
>
> **Errors received while creating a To Do entry**. If the background process cannot successfully create a To Do entry to report an object-specific error, the error is written to the batch run along with the error encountered while attempting to create the To Do entry.

> **Note**
>
> **System errors are not included**. To Do entries are not created for a system error, for example an error related to validation of input parameter. These errors are written to the batch run . Refer to Processing Errors for more information.

### To Dos Created by Background Processes for Specific Conditions

There are some system background processes that create a To Do entry when the process detects a specific condition that a user should investigate. For each background process, the To Do type is an input parameter to the process. The system provides To Do types for each base package background process that may create a To Do entry.

> **Note**
>
> **No Creation Process.** These To Do types do not need (and should not have) a **Creation Process** specified.

## To Do Entries Created By Algorithms

There are To Do entries that are created by algorithm types supplied with the base package. The system supplies a To Do Type for each of these To Do entries that you may use.

If you want to take advantage of these types of entries for system algorithm types, you must do the following:

- Create an algorithm:
  - This algorithm must reference the appropriate Algorithm Type.
  - These algorithms have a parameter of the To Do Type to be created. You should specify the To Do Type indicated in the table.
- Plug the algorithm into the respective control table.

## To Do Entries Created Manually

You must set up manual To Do entry types if you want your users to be able to create To Do entries online. Users may create a manual To Do entry as a reminder to themselves to


Copyright © 2026, Oracle and/or its affiliates.

complete a task. Online To Do entries may also be used like electronic help tickets in the system. For example, if a user is having a problem starting service, the user can create a To Do that describes the problem. The To Do can be assigned to a help resolution group that could either resolve the problem or send the To Do back to the initiating user with information describing how to resolve the problem.

If you want to take advantage of manual To Do entries, create a To Do type and specify the following information.

**On the Main tab:**

- Set the **To Do Type Usage** flag to **Manual**.
- Set the **Navigation Option** to **toDoEntryMaint** (To Do entry maintenance).
- Set the **Message Category** and **Message Number** to the message you want to be used for To Do entries of this type. The system will populate the message parameter with the Subject. To show only the subject in the To Do's message, use a message with "%1" as its text.

**On the Roles tab:**

- Specify the To Do roles that may be assigned to To Do entries of this type.
- Indicate the To Do role that should be defaulted when you create To Do entries of this type.

**On the Sort Keys tab:**

When a user adds a manual To Do entry, the system creates an entry with three sort key values. (Sort keys may be used on the To Do list page to sort the entries in a different order.) The To Do type should be set up to reference the sort keys as follows:

| Sequence | Description |
|---|---|
| 1 | Created by user ID |
| 2 | Created by user name |
| 3 | Subject |

We recommend that the keys have an **Ascending** sort order and that the Subject is made the default sort key.

> **Note**
>
> It is possible to define additional sort keys and use a To Do Post Processing algorithm to populate the values. In this case, the base sort keys defined above should still be defined.

**On the Drill Keys tab:**

When a user adds a manual To Do entry, it is created with a drill key value equal to the To Do entry's ID. When the user clicks the Go To button next to the message in the To Do list, the system uses the drill down application service (defined on the main tab) and the drill key to display the associated To Do entry.

The To Do type must be set up with a drill key that reference the To Do entry table and the To Do entry ID:

| Sequence | Table | Field |
|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

# Refining a To Do Entry's Details at Creation Time

When a To Do Entry is created, the key information provided to the system are the To Do Type and the drill key, which is the identifier of the object related to the To Do entry. With this information, system logic along with algorithms called at creation time can add details to the To Do entry or update important attributes so that the To Do can be worked on by the right team in the right priority.

The topics in this section will highlight the features provided to further refine a To Do Entry at creation time.

### Linking Additional Information to a To Do Entry

Additional information may be linked to a To Do entry using characteristics. This may include master data so that the To Do entry is associated with one or more specific main objects in the system. It may also include other information that is helpful to any user that may be working on the To Do entry.

For **manual** To Do entries, the valid characteristic types that may be linked to the To Do entry must be defined on the To Do Type for that To Do entry. When create a manual To Do entry, the user populates the related characteristics as part of creating the To Do.

For **automatic** To Do entries, the program that generates the To Do may link related data to the To Do using characteristics.

In addition, the system To Do Pre-creation plug-ins can be used to link additional characteristics. These algorithms would use the drill key, which is the related object for the To Do entry. Based on the drill key, it can look up other information and create appropriate characteristics. Your specific product may be providing one or more algorithms to determine key master data records associated with the related object. For example, Oracle Utilities Customer Care and Billing provides an algorithm that attempts to link a related person, account, premise, service agreement and service point for that drill key. Other edge applications provide a similar algorithm for their relevant entities.

Any information that your user may need to have to help work a To Do entry should be linked to the To Do via a characteristic. For example, if the Rate Class of an account is important or the amount of the previous bill will help determine the priority of the To Do entry, this information may need to be linked as a characteristic.

> **Note**
>
> With the volume of To Do entries in the system, it is recommended to carefully consider what information you are linking to the To Do as a characteristic. Users can drill down on details when investigating a To Do entry. Only very important and critical information that would help users in reviewing To Do entries in the To Do Management portal and make decisions about prioritization and assignments should be linked as characteristics.

Note, before you can set up these algorithms, you must define the characteristic types that you'll use to hold each of these entities. Also note that it is not necessary to define these characteristics as valid characteristic types on the To Do type.


Copyright © 2026, Oracle and/or its affiliates.

If your To Do entries reference characteristics that are related to your global context data, you may want to configure an alert algorithm to display an alert if a related entry is **Open** or **Assigned**. Refer to Installation Options for more information.

> **Note**
>
> Count of related To Do entries. There are some user interfaces that display a count of related To Do entries. This logic looks at foreign key characteristics linked to the To Do entry where the underlying object is a master or transaction object. It counts the distinct list of other non-closed To Do entries that reference the same characteristic(s).

### Adjusting the To Do Role

When creating a To Do entry, for a To Do type, there may be attributes of the To Do entry that require that a special set of users work on it. This is sometimes referred to as 'rerouting the To Do entry' to a different role. This section highlights ways to override the role based on conditions specific to a To Do entry.

The information in this section only applies to automatic To Do Entries created by an algorithm or background process. It does not apply to manually created To Do Entries.

Refer to To Do Entries Reference A Role for the hierarchy used to determine the role .

#### Reroute Based on To Do's Message

As described in Suppress the Creation of the To Do Entry, To Do entries created for a given To Do type may have different messages based on the specific condition for a record. You may decide that based on the message, the To Do should be routed to a specific Role that is more knowledgeable about a specific error condition. The same message override configuration that may be used for suppression of a To Do entry, may alternatively be used to assign a specific Role. For example, if certain bill segment errors are always resolved by the rate specialist team, you can indicate the role for that team on the To Do type, message override.

#### Reroute Based on a Condition

The To Do Pre-creation plug-ins may be used to assign a specific role based on some condition related to the To Do entry's specific data.

The product provides configuration on access group to define To Do assignment overrides. The override list defines a To Do type and To Do role. This is useful if your product associates access groups with specific master data for row level security purposes and you would also like to direct To Do entries related to this data to a specific role.

Oracle Utilities Customer Care and Billing also defines similar To Do assignment override configuration on CIS division and Account Management Group.

A pre-creation algorithm is responsible for checking if data linked to the To Do entry is related to a configuration object that supports an assignment override and then looks for an override for the To Do type. You can decide if this algorithm should be plugged in at the To Do Type level or the Installation level. The decision is based on how many To Do types you define overrides for. If it's a small number, then it makes more sense to plug the algorithm in for each To Do type that you define an override for. This ensure that the configuration is only checked for To Do entries that will have an override. However, if the number of To Do types that are eligible for overrides is larger or if the list changing or growing often, it may be better to plug the algorithm in at the Installation level to ensure that every To Do is checked for a possible override.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Note that your product may already provide a base delivered pre-creation algorithm that is checking for this override. If so, that can be used by your implementation. If not, your implementation can provide an appropriate algorithm.

#### To Do Pre-creation Plug-ins

The system provides two To Do Pre-creation plug-in spots that allow for functionality to add or update information on the To Do Entry that is about to be created. The two plug-in spots are To Do type and installation options. In general, the installation option To Do pre-creation algorithms would be used for logic that applies to many To Do types and the To Do type pre-creation algorithms would be used for logic that is specific to that To Do type.

If at least one To Do pre-creation algorithm is plugged in on the To Do type, only those algorithms are executed when creating the To Do entry and the installation option algorithms are ignored. The system provides a base delivered To Do type pre-creation algorithm that executes the Installation option To Do pre-creation algorithms. This allows you to control whether the installation algorithms should also be executed for a To Do type that has specific algorithms and if so, when.

The following points highlight functionality supported by both To Do pre-creation plug-in spots.

- Indicate that the To Do entry should be suppressed (not created). The expectation is that this type of algorithm is configured on the To Do type rather than at the installation options level, as this allows for more granular conditions to be checked for a given To Do entry. Refer to Suppress the Creation of the To Do Entry for more information.
- Linking context specific data to the To Do entry using characteristics. Refer to Linking Related Information to a To Do Entry for more information.
- Overriding the Role of a To Do entry based on specific configuration related to the To Do's characteristic data. Refer to Adjusting the To Do Role for more information.

#### Assigning a Priority

As described in The Priority of a To Do Entry, a To Do type may reference a Calculate Priority algorithm. If so, this algorithm is called as part of the steps to create a To Do entry. The algorithm may use the drill key or any of the related information linked to the To Do as described above to determine priority information.

## The Priority Of A To Do Entry

Some To Do entries may be more urgent to resolve than others. A To Do entry is associated with a priority level representing its relative processing order compared to other entries.

Priority level is assigned as follows:

- If one or more **Calculate Priority** plug-ins are defined on the To Do entry's type, the system calls them to determine the entry's priority. They are called initially when a To Do entry is created and each time it gets updated. You may want to use this method if an entry's priority is based on context or time-based factors. For example, when priority takes into consideration account specific attributes. The system also provides a batch process (`F1-TDCLP`) that calls the calculate priority algorithms "at will" for non-closed To Do entries. This is useful when the priority should be reassessed periodically based on factors external to the To Do entry's information. Refer to To Do Type for more information on


Copyright © 2026, Oracle and/or its affiliates.

priority calculation algorithms. When the priority is determined by one of the algorithms, a log entry is created indicating that the priority was calculated.

- If a priority value has not been determined by a plug-in, the system defaults a To Do entry's initial priority to the value specified on its type.

A user may manually override a To Do entry's priority at any time. When a user overrides the priority, a log entry is created indicating that the priority was overridden. Notice that once a To Do entry's priority is overridden, **Calculate Priority** plug-ins are no longer called so as to not override the value explicitly set by the user.

> **Note**
>
> The system does not use priority values to control order of assignment nor processing of To Do entries. Priority is available to assist your organization with supporting a business practice that ensures higher priority issues are worked on first.

## What Happens When You Work A To Do Entry

Many user interface tools allow the user to indicate that they will start working on a To Do entry. Refer to Working On A To Do for more details. When a user works on an entry, the user is transferred to the maintenance page associated with the entry (the To Do entry's drill key). The page to which the user is taken is configured on the To Do type.

It is also possible to configure the To Do type to launch a script when a user works on a To Do entry rather than taking the user to a specific page. The script can check conditions related to the To Do entry to determine a more appropriate page or could be configured to walk the user through the steps required to resolve the To Do entry.

A script is linked to a To Do type based on its message number using the To Do type's message overrides. Refer to Executing A Script When A To Do Is Selected for more information.

In certain pages, taking an action to work on a To Do entry automatically assigns the To Do entry to the user. For users in implementations that also take advantage of the assignment substatus values, the To Do entry is also updated to the **In Progress** substatus. Refer to The Lifecycle Of A To Do Entry for information about configuring the system to use the substatus values.

## Monitoring A To Do Entry

A To Do type may reference a To Do monitor algorithm. The algorithms are executed by the To Do monitor background process (F1-TDMON). The process executes the monitor algorithms for any non-complete To Do entry whose To Do type references at least one monitor algorithm.

Monitor algorithms may be useful if there are business requirements where a To Do could be automatically completed based on criteria that the algorithm can review. For example, an algorithm may be used to detect if the situation that caused the To Do Entry to be generated has been remedied in the meantime, allowing for the To Do Entry to be completed.

## To Do Entries Have Logs

Each To Do entry has a To Do log that maintains a record of the To Do's progress in the system. For example, the To Do log indicates when the To Do entry was created, when it was assigned to a user and to whom it was assigned, and when and by whom it was completed.


Copyright © 2026, Oracle and/or its affiliates.

Users can view the log to see who assigned them a particular To Do and whether any work has already been done on the To Do.

A log entry is created for all actions that can be taken on a To Do entry. Log entries are created for the following events:

- A To Do entry is created (either by the system or by a user)
- A To Do entry is completed (either by the system or by a user)
- A user takes an open To Do entry
- A supervisor assigns a To Do entry
- The To Do entry's status is updated and the event is not already covered by another log entry type
- A user forwards an entry to another user or role
- A user sends back a To Do to the user who forwarded it
- A user manually adds a log entry to record details about the To Do's progress
- A user manually overrides the To Do entry's priority
- The To Do entry's priority was updated as a result of a calculate priority algorithm.

**NOT_SUPPORTED**

For information about the contents of log entries for each of the events, refer to Log Entry Events.

## Implementing Additional To Do Entry Business Rules

If your business practice calls for additional validation rules or processing steps to take place after a To Do Entry is created or updated, you may want to take advantage of the **To Do Post Processing** plug-ins defined on To Do type.

For example, you may want to validate that To Do entries are only assigned to users with the proper skill levels needed to resolve them. Refer to `F1-VAL-SKILL` for a sample algorithm handling such validation.

## To Do Entries May Be Routed Out Of The System

A To Do type can be configured so that its entries are interfaced to another system.

For example, a given To Do type can be configured to create an Email message whenever a new To Do entry is created. The following points describe how to do this:

- Define the name of the background process responsible for interfacing the new To Do entries to another system on the respective To Do type. The base package contains a batch process called `F1-TDEER` that can be used for most situations. This batch process invokes the **External Routing** algorithms defined on each entry's To Do type.
- Plug in an appropriate **External Routing** algorithm on the respective To Do type. The logic in this type of algorithm performs the interface efforts for a specific To Do entry. For example, if an Email message should be created for a To Do entry, the logic in the algorithm would compose and send the Email message(s) for a specific To Do entry.


Copyright © 2026, Oracle and/or its affiliates.

## To Do Information May Be Formatted By An Algorithm

A **To Do Information** algorithm may be plugged in on the installation record to format the standard To Do information that appears throughout the system. This algorithm may be further overridden by a corresponding plug-in on the To Do Type.

## Periodically Purging To Do Entries

The To Do Entry is configured for Information Lifecycle Management . This is the recommended tool for managing older records.

A purging batch process is also supplied `F1-TDPG`. However, it has some limitations that should be considered. The purging process does not perform any referential integrity. As such, if you are proactively purging records that are not very old, they could be referenced by other objects that are still active.

The process purges **Completed** To Do entries. This background process offers you the following choices:

- You can purge all To Do entries older than a given number of days.
- You can purge To Do entries for a specific list of To Do types that are older than a given number of days.
- You can purge all To Do entries except for a specific list of To Do types that are older than a given number of days.

> **Note**
>
> **Different retention periods for different types of To Do entries.** Keep in mind that the purge program allows you to retain different types of entries for different periods of time.

# Setting Up To Do Options

The topics in this section describe how to set up To Do management options.

- Installation Options
- Messages
- On Demand Assignment Configuration
- Defining To Do Roles
- Defining To Do Types
- Implementing The To Do Entries

## Installation Options

There are a number of installation options that may be configured to govern various aspects of To Do processing.

- A **To Do Information** algorithm may be plugged in on the Installation record. Refer to To Do Entries My be Formatted By An Algorithm for more information.


Copyright © 2026, Oracle and/or its affiliates.

- A **To Do Pre-creation** algorithm may be plugged in on the Installation record to set additional information for a To Do entry before it is created. Refer to To Do Pre-creation Plug-ins for more information.
- A **Next To Do Assignment** algorithm must be plugged into the Installation record if your organization opts to use the on demand assignment feature.
- If your To Do entries reference characteristics related to your global context data and your product supports sidebar alerts generated by algorithms, you may want configure an algorithm to display an alert if an entry is **Open** or **Assigned** for the data currently in context. Refer to your product's documentation to determine if these types of alerts are supported.

# Messages

You need only set up new messages if you use algorithms to create To Do entries or prefer different messages than those associated with the base package's To Do types.

# On Demand Assignment Configuration

The base package provides configuration related to allowing your users to click the "Assign me a To Do" button on the Current To Do Zone on the sidebar. If your organization plans to use this feature, be sure that users are granted the **Get Next Assignment** the Current To Do Zone application service (FWLCTDEP).

**Next Assignment Plug-in**

In order for the user to get an assignment on demand, a Next Assignment algorithm needs to be plugged in on installation options. The algorithm is responsible for determining the next To Do entry the user should work on.

The product provides an algorithm out of the box (`F1-NEXT-TDNS`) that can assign work solely based on priority. Another algorithm (`F1-NEXT-TDSK`) may be used to only consider To Do entries the user has the proper skills to resolve. Your implementation may provide a different algorithm based on your own business rules.

**Skills**

To use the algorithm that attempts to match a user's skills to the To Do Type's skills, the product provides additional configuration that is needed.

- Define skills. You need to define one or more measurable skills so that you can assign a skill level to both the To Do Types and to your users. The algorithm expects you to use predefined characteristic types, one for each different type of skill needed. Define the characteristic values as numeric, where the higher the number, the higher the experience / skill level. The characteristic type should configure both the To Do Type and the User characteristic entities.
- Define message overrides, if applicable. If your To Do Type is used for exceptions, you may refine the skill needed for a given message. Do this by using an ad-hoc characteristic on the
- Update your To Do types with the characteristic types that indicate the skills needed and the minimum skill level for each.
- Skill requirements may also be overridden on a message level. The algorithm assumes an ad-hoc characteristic on To Do Type is used to capture skill override for a given message. The ad-hoc value format is as follows: msg=12,12011 skill=BILL-EXC level=1 where msg is the message number, skill is the characteristic type defining the type of skill and level is the minimum skill level.


Copyright © 2026, Oracle and/or its affiliates.

- Update your users that work on To Dos to indicate all their capable skills and skill levels. Create a characteristic entry for each skill defined and use the characteristic value to indicate the user's skill level. The algorithm provided will ensure that the user's skill is equal to or higher than the required skill for the To Do Type.
- Register the characteristic types you are using for skills so that the algorithm knows which ones to evaluate. Go to Feature Configuration and find (or create) the feature with the **To Do Options** feature configuration type. Define an entry with the option type **Skill** for each characteristic type defined.
- Register the characteristic type you are using to indicate the Override Skills for a message level override on the same feature option. Define an entry with the option type **Override Skill** for the characteristic type.

The base **To Do Post Processing** To Do Type algorithm `F1-VAL-SKILL` validates that a user has the necessary skill levels required to work on a given To Do entry.

# Defining To Do Roles

This section describes the control table used to maintain To Do roles.

- To Do Role - Main
- To Do Role - To Do Types

## To Do Role - Main

The **Main** page is used to define basic information about a To Do role.

To maintain this information, select **Admin**, and then **General**, and then **To Do Role**.

**Description of Page**

Enter a unique **To Do Role** and **Description** for the To Do role.

The grid contains the ID of each **User** that may view and work on entries assigned to this role. The First Name and Last Name associated with the user is displayed adjacent.

> **Note**
>
> **System Default Role**. The system supplies a default role `F1_DFLT` linked to each system To Do type. This is done so that To Do functionality may be tested prior to the creation of appropriate business oriented To Do roles.

In addition, various "type" objects or algorithms may reference a To Do role to use when creating a To Do for a given business scenario. This is dependent on your specific product.

## To Do Role - To Do Types

The **To Do Types** page defines the To Do types that may be viewed and worked on by users belonging to a given To Do role.

To maintain this information, select **Admin**, and then **General**, and then **To Do Role** and navigate to the **To Do Types** page.

**Description of Page**

Enter the ID of each **To Do Type** whose entries may be viewed and worked on by the role.


Copyright © 2026, Oracle and/or its affiliates.

**Use As Default** is a display-only field that indicates if the role is assigned to newly created entries of this type. You may define the default role for a given To Do type on the To Do Type maintenance page.

> ⚠ **Caution**
>
> If you remove a To Do type where this role is the default, you must define a new role as the default for the To Do type. You do this on the To Do Type maintenance page.

# Defining To Do Types

Use this page to define basic information about To Do types.

You can access the portal from the **Admin**, and then **General**, and then **To Do Type**. You are brought to a query portal with options for searching for a specific To Do Type. Once a To Do Type has been selected you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! If you introduce a To Do type, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **To Do Type.** This zone provides information about the selected To Do Type.
  - The **drill keys** used to navigate to the To Do entry's related page are defined on To Do Type.

    > ⚠ **Caution**
    >
    > Do not change this information unless you are positive that the process / algorithm that creates entries of a given type stores this information on the entries.

  - **Sort Keys** define attributes that may be used to sort To Do entries of this type. Note that this information is captured as a string and is meant for sorting purposes. The system does not capture data type information to be able to format the information for display. For example, dates are not going to display as per the user's display profile configuration.
    - The To Do Information zone on the To Do Management portal provides the ability to search for To Do entries using sort keys and includes up to five sort key values as columns in the results, which may be used for sorting as required.
    - The To Do list page includes an 'Extra Information' option to allow To Do entries to be sorted by a selected sort key. Note the default sort key for this type of sorting is also defined on To Do type.
  - Refer to To Do Entries Reference A Role for more information about **roles** and To Do entries.


Copyright © 2026, Oracle and/or its affiliates.

- Refer to The Priority Of A To Do Entry for more information about **priority** related setup.
- Refer to To Do Entries Created By Background Processes and To Do Entries May Be Routed Out Of The System for more information about associating a To Do Type with **batch processes**.

- **To Do Type Characteristics.** This zone lists characteristic types and values common to all To Do entries of this type.

The following zones may appear as part of the portal's **To Do Characteristics** tab page.

- **To Do Characteristics.** This zone is used to maintain the valid list of characteristics that can be defined for To Do entries of this type. The characteristic types for characteristics that are linked to the To Do entry as a result of a pre-creation algorithm do not need to be defined here.

The following zones may appear as part of the portal's **Override Messages** tab page.

- **Override Messages.** If you want To Do entries that reference a given message category / number to be routed to a specific To Do role or suppressed altogether or if you want to associate a script to a given message category / number, use this zone to specify these rules.
  - Refer to Suppress the Creation of the To Do Entry for more information on suppressing To Do creation.
  - Refer to To Do Entries Reference A Role and Adjusting the To Do Role for more information about overriding roles.
  - Refer to What Happens When You Work A To Do Entry for more information on user guidance.

The following zones may appear as part of the portal's **Algorithms** tab page.

- **Algorithms.** This zone is used to maintain the algorithms associated with the selected To Do Type.

#### Actions

The page supports the following special actions:

- **Create an Algorithm.** Use this action, on the Algorithms tab to creates a script based algorithm related to the selected To Do Type. You are prompted for information regarding the system event, the name, description, etc. Once all the information is provided, the system creates an algorithm type, a script, an algorithm and links the algorithm to the To Do Type. You may then navigate to the script to adjust it as needed.
- To Do Type Algorithms

## To Do Type Algorithms

The following table describes all the system events that control important To Do functions. If you haven't already done so, you must set up the appropriate algorithms in your system.


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
| --- | --- | --- |
| **Calculate Priority** | Optional | Algorithms of this type may be used to calculate a To Do entry's priority. Refer to The Priority of a To Do Entry for more information on when this plug-in is called.<br>Note that it is not the responsibility of the algorithms to actually update the To Do entry with the calculated priority value but rather only return the calculated value. The system carries out the update as necessary.<br>If more than one algorithm is plugged-in the system calls them one by one until the first to return a calculated priority. |
| **External Routing** | Optional | Algorithms of this type may be used to route a To Do entry to an external system.<br>The base package `F1-TDEER` background process invokes the algorithms for every To Do entry that its type references the process as the **Routing Process** and that the entry was not already routed. The background process marks an entry as routed by updating it with the batch control's current run number.<br>If more than one algorithm is plugged-in the batch process calls them one by one until the first to indicate the To Do entry was routed. |
| **To Do Information** | Optional | We use the term "To Do information" to describe the basic information that appears throughout the system to describe a To Do entry. The data that appears in "To Do information" is constructed using this algorithm.<br>Plug an algorithm into this spot to override the "To Do information" algorithm on installation options or the system default "To Do information" if no such algorithm is defined on installation options. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| **To Do Monitor** | Optional | Algorithms of this type are executed by the To Do Monitor background process and may be used to periodically review a To Do entry and perform actions, if needed. Refer to Monitoring a To Do Entry for more information. |
| **To Do Post-Processing** | Optional | Algorithms of this type may be used to validate and/or further process a To Do entry that has been added or updated. |
| **To Do Pre-Creation** | Optional | Algorithms of this type are called when a To Do entry is being added. Refer to Refining a To Do Entry's Details at Creation Time for more information. To Do Type pre-creation algorithms override the Installation level pre-creation algorithms. |

## Implementing The To Do Entries

To enable the To Do entries visible in the To Do Type page, you must configure the system as follows:

- Define the To Do roles associated with each To Do type and link the appropriate users to them. Once you have defined the roles appropriate for your organization's To Do types, remove the reference to this system default role `F1_DFLT`. Refer to To Do Entries Reference A Role for more information.
- For any To Do Type that is provided for a specific background process, the To Do simply needs to reference the appropriate Creation Background Process. When the background process is scheduled, To Dos are created based on the logic of the related background process. This applies to To Dos Created for Object-Specific Error Conditions and Dedicated To Do Background Processes.
- For any To Do Type that is provided for creation by an algorithm or other process, there may be configuration required to populate that To Do type as an algorithm parameter or as an attribute on a control table.

> **Note**
>
> Refer to the description of the To Do type for more information.

# Background Processes

This chapter covers various topics related to background processes. Besides providing an overview of background process functionality, the various tools available within the application to define, submit and monitor background processes are covered.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Your specific source application may have additional background process topics. Please refer to the documentation section that applies to your source application for more information.

- Understanding Background Processes
- Defining Batch Controls
- Defining File Integration Types
- Defining File Integration Records
- On-Line Batch Submission
- Tracking Batch Processes
- Requests

# Understanding Background Processes

This section describes various topics related to the background processes that perform many important functions throughout your product such as:

- Processing To Do Entries
- Monitor processes that select records in a given state to progress them to their next state in their lifecycle
- Processes that purge data
- Processes that extract data
- And many more...

- Background Processing Overview
- Batch Categories
- Parallel Background Processes
- Parameters Supplied To Background Processes
- Processing Errors
- Error Post-Processing Logic
- Post-Processing Logic
- Timed Batch Processes
- Monitor Background Processes
- Plug-in Driven Background Processes
- How to Re-extract Information
- How to Submit Batch Jobs
- How to Track Batch Jobs
- How to Restart Failed Jobs and Processes
- Assessing Level of Service
- Batch Resource Groups


Copyright © 2026, Oracle and/or its affiliates.

- Batch Control Options

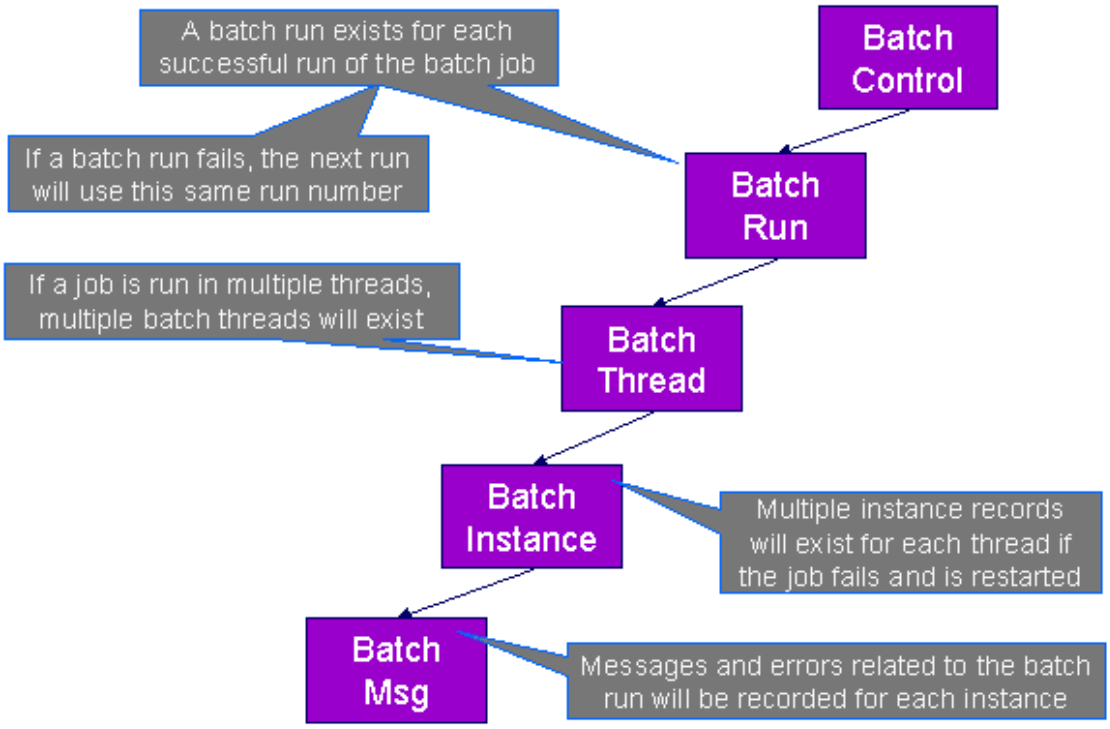
# Background Processing Overview

While the system relies on a scheduler to secure and execute its background processes, there are additional issues that you should be familiar with:

- Batch control records are used for the following purposes:
  - Define the code that executes the logic associated with the background process.
  - For processes that extract information, the batch control record defines the next batch number to be assigned to new records that are eligible for extraction. For example, the batch control record associated with the process that routes To Do entries to an external system defines the next batch number to be assigned to new To Do entries that are configured with this batch control. When this To Do external routing process next runs, it selects all To Do entries marked with the current batch number (and increments the next batch number).
  - The batch control record for each background process organizes audit information about the historical execution of the background process. The system uses this information to control the restart of failed processes. You can use this information to view error messages associated with failed runs.
  - Many processes have been designed to run in parallel in order to speed execution. For example, the process that applies updates for a migration data set import for CMA can be executed so that multiple "threads" are processing a different subset of records (and multiple threads can execute at the same time). Batch control records associated with this type of process organize audit information about each thread in every execution. The system uses this information to control the restart of failed threads. Refer to Parallel Background Processes for more information.
  - Some processes define extra parameters. These parameters are defined with the batch control. Default values may also be captured for each parameter. They will be used when the background process is submitted on-line.

The following diagram illustrates the relationships that exist for batch control records.


Copyright © 2026, Oracle and/or its affiliates.




Results of each batch run can be viewed using the Batch Run portal.

## Batch Categories

A batch process may be categorized for documentation purposes. The base values provided are as follows:

- **Ad Hoc**. Processes of this type are run on an ad hoc basis, only when needed. For example, if there is a process to do a mass cancel / correction of data, it would only be run when a situation occurs requiring this.
- **Extract**. Extract processes extract information that is interfaced out of the system. Processes of this type typically extract records marked with a given run number. If the requester of the process does not supply a specific run number, the system assumes that the latest run number should be extracted. If you need to re-extract an historical batch, you simply supply the respective run number when you request the batch process.
- **ILM**. Information Lifecycle Management jobs are crawler background processes that are associated with the ILM based storage solution.
- **Monitor**. Processes of this type are processes related to business objects with a lifecycle state that defines monitor algorithms. The monitor process selects records in a given state and executes its algorithms, which may cause the record to transition to another state or may trigger some other logic to occur. Using configuration, the monitor process may target only specific records. Refer to Monitoring Batch Processes for more information. Note that these types of background processes can be considered a subset of **Process What's Ready**
- **Process What's Ready**. Processes of this type create and update records that are "ready" for processing. The definition of "ready" differs for every process. For example, a payment upload process creates payments for every record that is **pending**. An overdue event monitor activates pending overdue events that have reached their trigger date.


Copyright © 2026, Oracle and/or its affiliates.

- **Purge**. Processes of this type are used to purge historical records from certain objects that generate a large number of entries and may become unwieldy over time.
- **To Do Entry**. Processes of this type are used to detect a given situation and create or complete a To Do Entry. Refer to To Do Entries Created by Background Processes for more information.
- **Upgrade**. Processes of this type are typically run only one time and are used to populate data in a table based on information that can be determined elsewhere in the system. These are processes that could have been done through an upgrade script, but perhaps due to data volume or timing, the upgrade logic warranted a separate stand-alone process.
- **Upload**. Processes of this type are used to upload data from a file into the system.
- The following categories are related to the data conversion / migration processes:
  - **Conversion**. Processes of this type are dedicated to converting or migrating data from external applications into the product.
  - **Object Validation**. Processes of this type are dedicated to validate data within objects for conversion or migration purposes.
  - **Referential Integrity**. Processes of this type are dedicated to validate referential integrity within objects for conversion or migration purposes.

> **Note**
>
> Additional categories may be introduced by your specific product.

## Parallel Background Processes

Many processes have been designed to run in parallel in order to speed execution. This is referred to as running the process with multiple “threads”.

The Submission Details section on the Batch Control page indicates whether the process is multi-threaded or not.

The system provides the following strategies for distributing the data to the multiple threads.

- **Thread Level SQL Select** (THDS). This strategy is sometimes referred to as the “thread iterator” strategy. In this strategy, the batch job uses the primary key to figure out how to evenly distribute key ranges to each thread. Each thread is then responsible for selecting the records. In this strategy, the threads should also re-select the data periodically to release the cursor, which aids in performance. Note that this strategy is preferred but may only be used under the following conditions:
  - The data from only one maintenance object is being processed.
  - The primary key for the maintenance object is a single, numeric system generated key.

  > **Note**
  >
  > Parameters may be used to override the low and high id. Refer to Parameters Supplied to Background Processes for more information.
- **Key Ranges** (KEYS). This strategy is similar to the **Thread Level SQL Select** strategy, and follows the same maintenance object and prime key conditions, but uses actual key values to calculate thread ranges. Such strategy may be needed in situations where keys


Copyright © 2026, Oracle and/or its affiliates.

do not evenly distribute across threads, resulting in uneven thread completion times. For example, conversion entities in the staging schema have legacy keys, which are typically not evenly assigned. This impacts the Object Validation and XML resolution batch processes that thread by legacy keys.

> **Note**
>
> This strategy should not be widely used and as such is only supported by the Monitor and Ad-hoc Plug-in Driven batch programs.

- **Job Level SQL Select** (JOBS). This strategy is sometimes referred to as the “standard commit” strategy. In this strategy, the keys for the records to be processed by the batch job are all selected first and stored in a temporary table. The batch job then supplies each thread with a range of keys that it should process. This strategy is used if multiple maintenance objects are being processed by the batch job; if the primary key of the maintenance object has multiple parts or if the primary key is non-numeric.

The multi-threading logic relies on the fact that primary keys for master and transaction data are typically system generated random keys. In addition, if the data is partitioned, it is expected to be partitioned based on the primary key.

> **Note**
>
> The detailed description in the metadata for each batch control provided with the system should indicate if it may be run in parallel. Note that the strategy used is not typically indicated in the detailed description.

> **Note**
>
> **Overriding the thread ranges**. Your implementation has the ability to override the thread ranges if certain data in your system takes longer to process. For example, imagine you have a single account in Oracle Utilities Customer Care and Billing that has thousands of service agreements (maybe the account for a large corporation or a major city). You may want to set up the thread ranges to put this large account into its own thread and distribute the other accounts to the other threads. To do this, you should create the appropriate batch thread records ahead of time in a status of **Thread Ready** ( `50`) with the key ranges pre-populated. Note that the base product does not provide the ability to add batch thread records online. If you are interested in more information about this technique, contact Customer Support.

- Optimal Thread Count

## Optimal Thread Count

Running a background process in multiple threads is almost always faster than running it in a single thread. The trick is determining the number of threads that is optimal for each process.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> A good rule of thumb is to have one thread for every 100 MHz of application server CPU available. For example if you have four 450 MHz processors available on your application server, you can start with 18 threads to begin your testing: (450 * 4) / 100 = 18.

This is a rule of thumb because each process is different and is dependent on the data in your database. Also, your hardware configuration (i.e., number of processors, speed of your disk drives, speed of the network between the database server and the application server) has an impact on the optimal number of threads. Please follow these guidelines to determine the optimal number of threads for each background process:

- Execute the background process using the number of threads dictated by the rule of thumb (described above). During this execution, monitor the utilization percentage of your application server, database server and network traffic.
- If you find that your database server has hit 100% utilization, but your application server hasn't one of the following is probably occurring:
  - There may be a problematic SQL statement executing during the process. You must capture a database trace to identify the problem SQL.
  - It is also possible that your commit frequency may be too large. Commit frequency is a parameter supplied to every background process. If it is too large, the database's hold queues can start swapping. Refer to Parameters Supplied to Background Processes for more information about this parameter.
- It is normal if you find that your application server has hit 100% utilization but your database server has not. This is normal because, in general, all processes are CPU bound and not IO bound. At this point, you should decrease the number of threads until just under 100% of the application server utilization is achieved. And this will be the optimal number of threads required for this background process.
- If you find that your application server has not hit 100% utilization, you should increase the number of threads until you achieve just under 100% utilization on the application server. And remember, the application server should achieve 100% utilization before the database server reaches 100% utilization. If this proves not to be true, something is probably wrong with an SQL statement and you must capture an SQL trace to determine the culprit.

Another way to achieve similar results is to start out with a small number of threads and increase the number of threads until you have maximized throughput. The definition of "throughput" may differ for each process but can be generalized as a simple count of the records processed in the batch run. For example, in the Billing background process in Oracle Utilities Customer Care and Billing, throughput is the number of bills processed per minute. If you opt to use this method, we recommend you graph a curve of throughput vs. number of threads. The graph should display a curve that is steep at first but then flattens as more threads are added. Eventually adding more threads will cause the throughput to decline. Through this type of analysis you can determine the optimum number of threads to execute for any given process.

# Parameters Supplied To Background Processes

This section describes the various types of parameters that are supplied to background processes.


Copyright © 2026, Oracle and/or its affiliates.

### General Parameters

The following information is passed to every background process.

- **Batch code.** Batch code is the unique identifier of the background process.
- **Batch thread number.** Thread number is only used for background processes that can be run in multiple parallel threads. It contains the relative thread number of the process. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives its relative thread number (1 through 20). Refer to Optimal Thread Count for Parallel Background Processes for more information.
- **Batch thread count.** Thread count is only used for background processes that can be run in multiple parallel threads. It contains the total number of parallel threads that have been scheduled. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives a thread count of 20. Refer to Optimal Thread Count for Parallel Background Processes for more information.
- **Batch rerun number.** Rerun number is only used for background processes that download information that belongs to given run number. It should only be supplied if you need to download an historical run (rather than the latest run).
- **Batch business date.** Business date is only used for background processes that use the current date in their processing. For example, a billing process may use the business date to determine which bill cycles should be downloaded. If this parameter is left blank, the system date is used. If supplied, this date must be in the format YYYY-MM-DD. Note: this parameter is only used during QA to test how processes behave over time.

  > **Note**
  >
  > There may be a slight lag in submission time for batch jobs submitted online. If the desired execution date/time is close to midnight, it is possible that your batch job will run on the day after you submit it. If you have left the business date blank in this case, keep in mind that your business date would be set to the day after you submit the job.

  > **Note**
  >
  > When rerunning a batch for a given run number, the system uses the batch business date of the original run.

- **Override maximum records between commits.** This parameter is optional and overrides each background process's Standard Commit. You would reduce this value, for example, if you were submitting a job during the day and you wanted more frequent commits to release held resources. You might want to increase this value when a background process is executed at night (or weekends) and you have a lot of memory on your servers.
- **Override maximum minutes between cursor re-initiation**. This parameter is optional and overrides each background process's Standard Cursor Re-Initiation Minutes. You would reduce these values, for example, if you were submitting a job during the day and you wanted more frequent commits to release held resources (or more frequent cursor initiations). You might want to increase these values when a background process is executed at night (or weekends) and you have a lot of memory on your servers.
- **User ID.** Please be aware of the following in respect of user ID:


Copyright © 2026, Oracle and/or its affiliates.

- Both the user submitting the job and the user ID recorded on the batch submission should have access to the application service for the batch control that secures execution.
- Any batch process that stamps a user ID on a record it creates or updates uses this user ID in applicable processing.
- This user ID's display profile controls how dates and currency values are formatted in messages.
- When submitting a batch job online, the batch user, language and email address are defaulted to the details of the user submitting the job. Only users that are granted the User Override access mode on the Batch Job Submission application service can override the batch user ID.
- If the submission user has user override privileges, they have the ability to retain the original batch user details when duplicating or rerunning a batch job from online submission. If the 'Run as original user' box is checked, the batch user, language and email address from the original batch run will be copied to the new batch job. If not, the user details are defaulted from the submission user.

- **Password.** Password is not currently used.
- **Language Code.** Language code is used to access language-specific control table values. For example, error messages are presented in this language code.
- **Trace Program Start**, **Trace Program Exit**, **Trace SQL** and **Trace Output**. These switches are only used during QA and benchmarking. If trace program start is checked, a message is displayed whenever a program is started. If trace program exit is checked, a message is displayed whenever a program is exited. If trace SQL is checked, a message is displayed whenever an SQL statement is executed. If trace output is checked, special messages formatted by the background process are written.

> **Note**
>
> The information displayed when the trace output switch is turned on depends on each background process. It is possible that a background process displays no special information for this switch.

### Common Additional Parameters

Each batch control supports the definition of additional parameters. There are some additional parameters that are common to all batch processes or common to a specific type of batch process. The batch control should be delivered with the appropriate additional parameters. However, when new additional parameters are introduced, existing batch controls may not be updated with the new additional parameter.

The following table highlights the common parameters that may be linked to a batch control. Note that for batch parameters, although there is a sequence number that controls the displayed order of the parameter, the batch process does not use the sequence to identify a particular parameter but rather uses the parameter name. In some cases multiple parameter names are supported (a ‘camel case’ version and an ‘all caps’ version).


Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Description | Additional Comments |
|---|---|---|
| MAX-ERRORS / **maxErrors** | Each of the batch processes has, as part of its run parameters, a preset constant that determines how many errors that batch process may encounter before it is required to abort the run. A user can override that constant using this parameter. | The input value must be an integer that is greater than or equal to zero. The maximum valid value for this parameter is 999,999,999,999,999. |
| DIST-THD-POOL | Each batch process executes in a thread pool. This parameter is only necessary if the batch process should execute in a different thread pool than the default thread pool. | The default thread pool name is **DEFAULT**. |
| emailMode | When the batch job is submitted with an associated email address, the default logic is to send an email when the job completes regardless of success or failure. Use this parameter to limit the email based on the status of the job when it ends. | Valid Values<br>• **ERROR** — send an email only when the job ends in Error status.<br>• **SUCCESS** — send an email only when the job ends in successfully.<br>• **ALL** — always send an email only when the job ends. (This is the default.) |
| The following parameters are only applicable to jobs that use the **Thread Level SQL Select** method of distributing work to threads as described in Parallel Background Processes. | | |
| overrideLowIdValue | Specifies a new low id to use in calculating the range for a thread. The framework by default assumes that the Id is between 0's (e.g. 000000000) and 9's (e.g. 9999999999), but this parameter will override the low value. | The parameter value can be an actual number or it can be set to **auto**. If **auto** is configured, it is set to the lowest current value on the database table associated with the background process. |
| overrideHighIdValue | Specifies a new high id to use in calculating the range for a thread. The framework by default assumes that the Id is between 0's (e.g. 000000000) and 9's (e.g. 9999999999), but this parameter will override the high value. | The parameter value can be an actual number or it can be set to **auto**. If **auto** is configured, it is set to the highest current value on the database table associated with the background process. |
| idRangeOverrideClass | Use this parameter to specify a custom class to do thread range calculation. During batch execution, this override class is instantiated and the setter methods called to initialize the Ids as required. The low and high getter methods are called to retrieve the high and low ids to be used for the run. | The class name specified must implement interface com.splwg.base.api.batch.BatchIdRangeOverride. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Description | Additional Comments |
|---|---|---|
| batchStrategy | Use this parameter to override a THDS batch strategy with either JOBS - Job-Level or KEYS - Key Ranges strategy. | This parameter is only supported by the Monitor and Ad-hoc Plug-in Driven batch programs. |
| The following parameters are only applicable to jobs that perform a single commit, for example for extract batch jobs. | | |
| numRecordsToFlush | This parameter defines how frequently to flush the Hibernate cache to prevent high heap consumption and Out Of Memory Errors. | |

### Specific Batch Parameters

Some background processes define additional parameters that are specific to their functionality. When a process receives additional parameters, they are defined and documented in the batch control entry in the application.

When submitting a batch job online, any default parameter values configured on the batch control record will be defaulted and may be overridden. Note that if the parameter value is configured to be Encrypted on the batch control configuration, the value will be shown encrypted.

- Indicating a File Path or File Name
- Encrypting and Signing Files
- Manifest File
- File Exchange through Oracle Cloud Storage

## Indicating a File Path or File Name

Some of the system background processes use extra parameters to indicate a File Path and/or File Name for an input file or an output file. For example, most extract processes use File Path and File Name parameter to indicate where to place the output file.

For the file path parameter, refer to Referencing URIs for several points related to populating the file path and the supported syntax.

For the file name parameter, there are extract processes that support defining a file name that ends in ".zip" or ".gz", which will automatically result in writing the file with zip or gzip compression (based on the suffix). Similarly, there are upload processes that support decompression of files using the appropriate method based on the file name ending in ".zip" or ".gz". If the extract or upload process supports compression / decompression, it should be indicated accordingly in the description of the File Name parameter. If a batch control doesn't explicitly indicate that it supports compression or decompression in its description, the assumption should be that it is not supported.

In addition, there are extract and import processes that support automatic signing, encryption and decryption of the file content using PGP standards. File names ending in .pgp will be parsed to determine whether encryption, signing, compression or a combination of the three applies. If the file has a .pgp extension and requires decryption or signature verification, a run time error will be issued if no encryption or signing key ring parameters are supplied.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> File name / extension validation. The system ensures that the file name for any file being uploaded does not contain special characters and uses only standard UTF-8 supported characters. In addition, the system validates that the file's extension is referenced in the base delivered allow list F1-FileExtensionAllowList, defined as a managed content record.

> **Note**
>
> Custom file extensions. If your implementation has defined custom file extensions that are not found in the base product delivered allow list, you may create a custom managed content record to define your additional valid extensions. To do this, duplicate the F1-FileExtensionAllowList record and give it the name CM-FileExtensionAllowList. Change the description as desired. Remove the base delivered extensions and add your valid extensions using the <extension></extension> tags.

## Encrypting and Signing Files

Some files created by data export related batch processes, may need to be encrypted and/or digitally signed for security purposes. Similarly, batch upload processes may need to verify the signature and/or decrypt the content of the file being imported.

New batch parameters control whether encryption or digital signing applies to files extracted or uploaded by the process. Both encryption and signing use Pretty Good Privacy (PGP) standards.

Processes that include a file name parameter, such as those using the plug-in driven extract template, support a file name extension of '.pgp'. Files ending with this extension are encrypted and/or signed using the encryption key parameters described in the next section.

Processes that have an internally constructed format for the file name, such as those using the generalized initial export template, perform encryption and/or signing according to whether the encryption parameters are supplied. An extension of '.pgp' is added to the constructed file name.

### Extract Process Encryption Parameters

| Parameter Name | Description | Additional Comments |
|---|---|---|
| `encryptExternalKey` | Use this key ring parameter to configure the external key provided by the recipient of the file. | If populated, the batch output file will be encrypted using a runtime generated key. The generated key will be encrypted using the active public key stored on the key ring. The recipient will use their private key to decrypt the generated key so the file may be decrypted. Only external public key rings may be used. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Description | Additional Comments |
|---|---|---|
| encryptSignatureKey | Use this key ring parameter to configure the internal key used to sign the file. | If populated, the batch output file will be signed using the active private key stored on the key ring. The assumption is that the corresponding public key has been shared with the recipient. Only internal file encrypting / signing key rings may be used. |

**Upload Process Encryption Parameters**

| Parameter Name | Description | Additional Comments |
|---|---|---|
| decryptInternalKey | Use this key ring parameter to configure the internal key to be used to decrypt the file to be uploaded. | If populated, the generated key received with the input file will be decrypted using the active private key stored on the key ring. The data is then decrypted using the generated key. Only internal file encrypting / signing key rings may be used. |
| decryptSignatureKey | Use this key ring parameter to configure the public signature key provided by the sender of the file. | If populated, the input file signature will be verified using the active public key stored on the key ring. Only external key rings may be used. |

You may use the Batch Control Query portal to find all batch controls supporting these options.

Refer to File Encryption and Signing Keys for more information on maintaining internal and external encryption keys.

## Manifest File

Batch processes that create files, for example data export related batch processes, may create one or more files in each processing thread. External applications that consume these files need to process them only when the entire batch process is successfully completed. To automate the consumption of such files, the external application may leverage a function to be triggered upon the creation of a manifest file to process all the files referenced by the manifest. The manifest file includes details such as content hash value, file size information etc that assist in confirming that file content is completed and has not been tampered with prior to processing the files.

The following sections provide additional information about manifest files.

### The Manifest Option Batch Parameter

Manifest file creation is supported on certain base product extract related batch controls only. These batch controls include the **Manifest Option** batch parameter that controls whether a manifest file should be created or not upon successful completion of the batch process. You may use the Batch Control Query portal to find all batch controls supporting this option.


Copyright © 2026, Oracle and/or its affiliates.

### Manifest File Information

Manifest file content is provided in JSON format and includes the following information:

- The version of the manifest details structure.
- The batch control code, run number and rerun number identifying the batch run producing the files
- Total number of files created by the batch run.
- Total number of records processed by the batch run.
- The manifest creation timestamp in ISO format.
- The previously completed batch run number, if any.
- The manifest file produced by the previously completed batch run, if any. This may be used to ensure files are processed in a proper order and assist in detecting potential situations of missing files.
- A list of files created by the batch run that includes the following details for each file: the file name, its size in bytes, and the SHA-256 hash value of the file's content.

The manifest file name is constructed as follows:

```
[batch control]_[batch run number]_[batch rerun number]_manifest.json
```

The following is an example of the manifest file structure.

```
{
  "version": 1,
  "batchCode": "HD-PDE",
  "batchRunNumber": 106,
  "batchRerunNumber": 0,
  "totalFiles": 3,
  "recordsProcessed": 10,
  "timeStamp": "2020-05-02T16:55:43.000353Z",
  "previousBatchRunNumber":"105"
  "previousManifestFileName":"HD-PDE_105_0_manifest.json",
  "fileDetails": [
    {
      "fileName": "SOME_FILE_NAME_001.json.gz",
      "fileSize": 1543,
      "hashAlgorithm": "SHA-256",
      "hashValue":
"ff7d4e466f93cadc4a18ab009080916baff631e03b728e2f2684c4a4f92ccbe1"
    },
    {
      "fileName": "SOME_FILE_NAME_002.json.gz",
      "fileSize": 1182,
      "hashAlgorithm": "SHA-256",
      "hashValue":
"78869e7ce971f28a70f5b35693a05726ba66109fecf492edb60291a62ae1ba9f"
    },
    {
      "fileName": "SOME_FILE_NAME_003.json.gz",
      "fileSize": 1182,
      "hashAlgorithm": "SHA-256",
```


Copyright © 2026, Oracle and/or its affiliates.

```
      "hashValue":
"78869e7ce971f28a70f5b35693a05726ba66109fecf492edb60291a62ae1ba9f"
    }
  ]
}{
```

#### Verifying The Manifest File

The **Verify Manifest File** web service may be used by an external system to validate that the manifest file it has received for a specific batch run is not corrupted or tampered with. The service returns the manifest file name and the hash value of its content for the requested batch run allowing the caller to compare their file details with the returned values.

## File Exchange through Oracle Cloud Storage

Oracle Utilities Cloud Services can exchange files bidirectionally with Opower Cloud Services and Energy and Water Cloud Services through an Oracle Utilities Cloud Service-managed Object Storage bucket. Specific base product integrations may be configured to write files to a cloud service managed bucket or to read information from a file found in a cloud service managed bucket. (Integrations that support these use cases document the capability directly in the description of the batch control delivered for each use case.)

> **Note**
>
> This option is only available in a cloud installation.

#### Sharing Customer Exported Data

Some data export related batch processes support an option that streamlines the process by which customer extracted data may be routed to and shared with Oracle so it can be uploaded to Oracle's data lake for analytic purposes.

Implementations may choose to share data with Oracle in addition to exporting data for their own purposes. For this use case, you could configure the batch control parameters as follows:

- Define the file path / file location parameter with the appropriate configuration for extracting the data to your location. Refer to Indicating a File Path or File Name for more information.
- The batch controls that support this include the **Copy to Oracle** batch parameter. Use this parameter to indicate that you want to also share the data with Oracle.

You may have a use case where you only want to extract the data to Oracle and you do not need to write this data to any extract location for your own use. For this use case, configure the batch control parameter for file path / file location with the mnemonic %ORACLE%. This mnemonic signals to the program to extract the data to the configured Oracle location only. There is no need to populate anything in the Copy to Oracle parameter.

# Processing Errors

When a background process detects an error, the error may or may not be related to a specific object that is being processed. For example, if the program finds an error during batch parameter validation, this error is not object-specific. However, if the program finds an error while processing a specific bill, this error is object-specific. The system reports errors in one of the following ways:

- Errors that are not object-specific are written to the error message log in the Batch Run.


Copyright © 2026, Oracle and/or its affiliates.

- Some batch processes create entries in an "exception table" for certain object-specific errors. For example, an error detected in the creation of a bill in Oracle Utilities Customer Care and Billing may be written to the bill exception table. If an error is written to an exception table, it does not appear in the batch run portal. For each exception table, there is an associated to do entry process that creates a To Do Entry for each error to allow a user to correct the problem on-line.
- For some background processes, errors that do not result in the creation of an exception record may instead generate a To Do entry directly. For these processes, if you wish the system to directly create a To Do entry, you must configure the To Do type appropriately. Refer to To Do entry for object-specific errors for information about configuring the To Do type. If the background process detects an object specific error and you have configured the system to create a To Do entry, the error is not written to the batch run. If you have configured your To Do type to not create To Do entries for certain errors, these errors are written to the Batch Run.

> **Note**
>
> **Some processes create exceptions and To Do entries**. It is possible for a background process to create entries in an exception table and create To Do entries directly, depending on the error. Consider batch billing in Oracle Utilities Customer Care and Billing; any conditions that cause a bill or bill segment to be created in **error** status result in a record added to the bill exception table or the bill segment exception table. However, any object-specific error that is not related to a specific bill or bill segment or any error that prevents a bill or bill segment from being created may result in a To Do entry for the object-specific error.

## Error Post-Processing Logic

The product supports executing one or more algorithms when a batch process encounters an error that causes execution to stop. This allows for some special processing to occur to handle the failure of the batch job. Algorithms for this plug-in spot receive the batch control, batch run number, batch processing business date, number of threads and the list of the ad hoc parameters of the batch job.

> **Note**
>
> This plug-in spot is available for all Java based batch programs. For programs of type **Java (converted)**, the individual batch programs need to explicitly support this plug-in spot. Unless otherwise noted, assume that a **Java (converted)** program does not support it.

The following are some examples of functionality that may be executed when a batch job fails:

- Another object or record that is monitoring the batch job may have its status updated to reflect the batch status.
- An outbound message service may be invoked to perform a task related to the failure.

Note that the units of work for all threads are committed prior to executing the error post-processing logic.


Copyright © 2026, Oracle and/or its affiliates.

## Post-Processing Logic

The product supports executing one or more algorithms after all the threads of a given batch job have completed. This allows for some special processing to occur at the end of a batch job. Algorithms for this plug-in spot receive the batch control, batch run number, batch processing business date, number of threads and the list of the ad hoc parameters of the batch job.

> **Note**
>
> This plug-in spot is available for all Java based batch programs. For programs of type **Java (converted)**, the individual batch programs need to explicitly support this plug-in spot. Unless otherwise noted, assume that a **Java (converted)** program does not support it.

The following are some examples of functionality that may be executed at the end of a batch job:

- Another dependent batch job can be kicked off. Note that this use case is only needed when the multiple dependent jobs are not part of a scheduler (which can also detect the successful end of one batch job so as to submit the next job).
- Statistics for the batch run may be analyzed and based on results, a To Do Entry may be sent to an administrator.
- If the current batch job is processing a large number of child records in multiple threads, a parent record could be updated to a different status or with some other audit information.

Note that the units of work for all threads are committed prior to executing the post-processing logic. The algorithm should perform standard error handling. If an error occurs in one of the post-processing algorithms, the overall batch job's status is set to Error so that it can be re-submitted to retry the logic in the finalize step.

## Timed Batch Processes

Most batch jobs are submitted via a batch scheduler. In the absence of a scheduler, a batch control may be configured as "timed" triggering the framework to monitor and schedule these batch jobs as defined by the timer interval. The timer interval defines the desired interval between starts (in seconds). The system schedules new batch runs at each interval if the last instance of the job has completed.

When configuring a batch control as "timed", other default information must be provided, including the User ID and Language to use for submitting the job and the email address for notification, if desired.

Timed batch controls also include an Active setting, allowing for an implementation to temporarily stop further executions of the batch job (but retain the other timer settings).

Timed jobs are controlled by the default threadpool and not by a scheduler. When the **DEFAULT** threadpoolworker starts it will start executing any job for a Batch Control configured as **Timed** with the Timer Active set to **Yes**. This is whether the batch daemon or batch server is enabled or not.


Copyright © 2026, Oracle and/or its affiliates.

# Monitor Background Processes

In many areas of the system, functionality is driven from business object configuration as a BO driven record progresses through its lifecycle. Refer to Business Object Lifecycle for details. As part of that functionality, it is possible that a background process, called a monitor batch process, is used to execute functionality for the record.

Monitor processes are designed to support parallel threads processing. When the number of records to process is unpredictable and can range from zero or few records to a many, submitting the batch with a fixed number of threads that fits the higher volumes means waste of threads with each run. Monitor processes are therefore designed to not create any batch run, and none of the thread level artifacts associated with a batch run, when there is no data to process. The batch job submission record is completed as usual but indicates that no batch run was needed.

> **Note**
>
> In an on-premise installation, monitor batch processes may be submitted via the command line, i.e. without creating a batch job submission record. In this situation, a monitor batch run will still be created even if there is no data to process because without a batch job record there is no way to communicate the reason for not creating a batch run.

The following sections further expand on concepts and features associated with monitor rule processing.

- Monitor Batch Template
- Deferred Requests Queue

## Monitor Batch Template

A single batch program is provided for the BO monitor functionality. The product also provides a batch control template (`F1-MNTPL`) that should be used as a starting point. Parameters are used to limit the records processed by maintenance object and other optional parameters that may further limit the records. The product typically provides at least one monitor batch control for each maintenance object that supports a configurable lifecycle on its business object.

This topic highlights the parameters supported by the monitor batch job. Not all parameters are applicable to all maintenance objects and therefore may not be configured on a given base monitor batch control.

| Parameter Name | Description | Comments |
|---|---|---|
| `maintenanceObject` | Maintenance Object | For most base delivered batch controls, this parameter is delivered already populated with the value of the maintenance object value. Note that it is supported to leave this value blank, at which point, the program will determine the maintenance object (objects) to process by looking for an MO that refers to this batch control record as an option. |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Description | Comments |
|---|---|---|
| isRestrictedByBatchCode | Restrict by Batch Code | Set this to **true** to indicate whether the process should only select records that explicitly refer to this batch control on its current BO state. This is also referred to as "deferred" mode. If set to **false**, the program includes all records that refer to the current batch control in its BO state and records that don't refer to any batch control in its current state (but monitor algorithms exist in the current state). This is commonly referred to as "periodic" mode. Note that if the value is not set at all, the program will determine whether to run it as "deferred" or "periodic" based on whether the batch code is configured on the MO option as a **State Monitor Process** (“deferred”) or a **Periodic Monitor Process**. |
| restrictToType | Restrict by Related Type | This parameter is only applicable to maintenance objects that have a related 'type' object and the maintenance object has configured an option indicating the field for the related type column. This parameter may be used to limit the processing to records that are in the indicated type. |
| restrictToBusinessObject | Restrict by Business Object | This parameter may be used to limit the processing to records that are in the indicated business object. |
| restrictToBOStatus | Restrict by Status | This parameter may be used to limit the processing to records that are in the indicated status. |
| restrictToDate | Restrict By Date | Enter a valid date on the record's primary table to limit processing to records that are have a value in this field that is on or before the batch business date. Optionally, enter "+" or "-" followed by a number to shift the comparison date to the batch business date plus or minus the given number. For example, entering REVIEW_DT -3 will retrieve all records whose review date is on or before the batch business date minus 3 days. |
| sampleRecordNumber | Sample Record Number | This is not a commonly used parameter. It is only applicable when the monitor is used for a business use case that supports processing a subset of the records during a testing phase. For example, if the process is validating a large number of records, it may be an option to only validate every 100 records to determine if there are repeated validation errors that may indicate a common problem that may be solved to fix many errors. |
| minimumRecordsPerThread | Minimum Records Per Thread | Enter a number to force the batch framework to dynamically adjust the number of threads used by the batch program, ensuring that threads are not created for a small number of records.<br>For example,<br>• If the batch is submitted with 100 threads, the parameter is set to 50 records per thread, and there are 1M records to process, then there is no change to the number of allocated threads.<br>• If the batch is submitted with 100 threads, the parameter is set to 50 records per thread, and there are 109 records to process, then the number of allocated threads is set to 3. |

Also note that when submitting a monitor process with multiple parallel threads, the program will use a **Thread Level SQL Select** strategy unless any of the following are true (in which case it will use the **Job Level SQL Select** strategy:


Copyright © 2026, Oracle and/or its affiliates.

- The input maintenance object is left blank and the program finds more than one maintenance object that refers to this batch control in its options.
- A single MO is applicable but it has a multi-part primary key.
- A single MO is applicable and it has a single primary key, but it is a user defined key instead of a system generated key.
- The sample record number parameter is populated.

In the rare situation when the **Thread Level SQL Select** (THDS) strategy should be used but actual prime keys are not evenly distributed you may override the strategy to use actual key ranges by setting the **Batch Strategy** general batch parameter to **Key Ranges** (KEYS).

## Deferred Requests Queue

There are use cases where an entity is set to a status that waits for a subsequent task to be performed asynchronously before it can progress in its lifecycle. Such task is implemented using monitor rules plugged into the entity's business object status definition along with a batch control. These monitor rules are not processed right away when the record enters that status but rather get deferred to when the associated batch runs next.

When the deferred task should be processed as soon as possible, a message can be added to the Deferred Requests Queue to perform the task near real-time. This mechanism offers a managed queue of messages, each representing a request to perform the deferred monitor rules of a given entity, similar to how these rules are processed in batch.

> **Note**
>
> This mechanism is only available in a cloud installation.

> **Caution**
>
> This feature should only be used to support business flows with time dependent events that require near real-time processing. All other asynchronous flows should remain deferred to batch processing.

The following sections describe concepts and features of the near real-time processing mechanism.

### Requesting Near Real-time Processing

When an entity enters a status that is associated with time dependent deferred monitor rules, a request can be made to process these rules as soon as possible after the current transaction is committed.

The product supports the following options for requesting near real-time monitoring of an entity:

- To always send the entity for near real -time processing when it reaches a specific status, add the 'Add a Deferred Request' (`F1-DFREQADDN`) 'Enter' algorithm to the respective business object status definition. Refer to the algorithm detailed description for more information.
- You may use the **DeferredRequestAPI** function in a Groovy based algorithm to implement custom rules around when and how to add such request to the queue.


Copyright © 2026, Oracle and/or its affiliates.

- When applicable, the Generic Data Synchronization functionality also supports an option for near real-time processing that uses the Deferred Requests queue.

### Exception Handling

When the deferred request is successfully processed by the real-time queue agent, the related entity is updated as per its business object rules. However, if the request keeps failing after several retries, the queue agent discards the request and the related entity stays in its current status, waiting to be picked up by the corresponding deferred monitor batch. At that point, if the error still persists the batch reports the issue as usual.

> **⚠ Warning**
>
> Do not remove the deferred batch control from the business object status definitions. These flows must remain deferred to batch as designed for data integrity reasons.

When applicable, sending entities to be processed near real-time allows the corresponding monitor batch to be scheduled less often as it no longer needs to meet near real-time frequency requirements. With most of the records successfully processed by the near real-time mechanism, the monitor batch is mainly used to handle exceptions.

### Correlated Requests

By default, messages in the queue are processed in no particular order. In situations where multiple messages update the same resource there may be a need to process them serially to prevent concurrency errors.

When adding a message to the queue, the message can optionally reference the main entity it updates for the purpose of correlating it with other messages that update the same entity. Messages explicitly referencing the same main entity are processed serially.

### Apply Polling Delay

When an entity's current status is associated with monitor rules that wait for a specific condition to be met in order to determine the next status the entity should transition to, it is not enough to perform these monitor rules once. The condition may not be met initially but could be met within a short period of time. In this situation, the deferred request placed in the queue could indicate that it requires polling delay.

When polling delay is requested, the deferred request message in the queue is processed as follows:

- When the message is added to the queue the current status of the related entity is noted.
- After successfully processing the message, if the related entity's status remains unchanged, the queue agent completes the current message and adds another message to the queue along with a short time delay, i.e. sets the entity to be processed again later.
- This process repeats itself for a limited number of attempts. With each attempt the delay in seconds increases.
- If the related entity's status remains unchanged after all attempts then the message is discarded with no exception. The related entity would be monitored as usual by the next scheduled batch run.


Copyright © 2026, Oracle and/or its affiliates.

# Plug-in Driven Background Processes

Although the product is delivered with a rich library of background processes, implementations may have business requirements that require new processes to be introduced. It is possible for an implementation to write a background process from scratch using a base process as a template. However, the product also provides base background processes that call algorithms to do the work that is needed. These are called plug-in driven background processes. There are three major types of plug-in driven batch processes:

- Processes that act on records that are stored in the database in the system. These types of processes require SQL to select the records along with the logic to process the records.
- Processes that extract information to a file. These types of processes require SQL to select the records along with the logic to select the information and return it such that the process can write the information based on various settings.
- Processes that import data from a file and store new records in the system as a result. These types of processes require an algorithm that is able to map the data from the file to appropriate new records in the system

The subsequent sections provide more detail about the three types of plug-in driven background processes.

- Ad hoc Processes
- Extract Record Processing
- Uploading Records

# Ad hoc Processes

A process that retrieves records in the system and does some action on them requires an algorithm to select the records to be processed and another algorithm to process the records. We refer to this type of process as an Ad hoc Process. The base batch control Plug-in Driven Generic Template (`F1-PDBG`) may be used as a template.

The base processes implement standard background process functionality including parallel background process logic and the ability to create To Do entries for errors. This allows for an implementation to take advantage of the pre-built support and provide plug-ins that include the logic that is unique to the specific use case.

The following sections provide more information about the functionality for ad hoc processes. Note that extract processes share some of the same functionality. Refer to Extract Record Processing for additional information specific to extract processes.

### Select Records Algorithm

The first important algorithm to design when implementing a plug-in driven batch process is the Select Records algorithm, plugged in on the batch control page. This algorithm type must define the first parameter as the SQL. The batch job will directly access the SQL parameter value in the metadata (rather than invoking the algorithm). All other parameters are available for the algorithm to use for its own logic.

In addition, when invoking the algorithm, it must return the strategy to use (**Thread Level SQL Select** or **Job Level SQL Select**. Refer to parallel background processes for more information about the two strategies and when to use each. When choosing the **Thread Level SQL Select** strategy, the algorithm should return the name of the primary key in the Key Field parameter. In addition, the SQL should include a **BETWEEN** clause that includes the bind variables for the low and high ID for the ranges. See below for the bind variable syntax.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In the rare situation when the **Thread Level SQL Select** (THDS) strategy should be used but actual prime keys are not evenly distributed the strategy may be overridden at batch submission time to use actual key ranges by setting the **Batch Strategy** general batch parameter to **Key Ranges** (KEYS).

If the SQL statement includes variables that are determined at execution time, it must use bind parameters. Bind parameters are referenced in the SQL statement using a colon and a parameter name (for example `:parameter`). There are some variables provided by the system that are populated by the batch job at execution time. These have `f1_` as its prefix.

The system supports the following pre-defined bind parameters:

- **:f1_lowID** and **:f1_highID** - these should be used in the **BETWEEN** clause for the **Thread Level SQL Select** strategy. They should be selecting data from the Key Field that is returned as part of the Select Records algorithm. The batch job will substitute the appropriate ID range as required.

  > **Note**
  >
  > The table where this key field is selected from should be the first table in the FROM clause.

- **:f1_batchCode** and **:f1_batchNumber** - these are common attributes of the batch control that are referenced on a record for selection purposes. Note that the batch run number is set according to whether the batch job is a re-run of a previous run or not.
- **:f1_businessDate** - the batch job will populate the input batch business date, if populated otherwise the current date.

> **Note**
>
> The system supports both "f1." and "f1_" as a prefix for the bind variables. However, the "f1." prefix will result in an error if the SQL security property setting is turned on. As such, the underscore syntax is recommended. The period syntax remains for backwards compatibility.

For any other custom parameters, the Select Records algorithm may return one or more sets of field name / variable name / value where the variable name matches a bind variable in the SQL. The field name provides information about the length that assists the SQL binding logic to properly substitute the values. Note that the variable name cannot start with `f1.` as its prefix. The batch job will use the value returned by the algorithm to set the bind parameter in the SQL statement. Also note that the logic binds all variables as a string, so for date variables the SQL must include the "TO_DATE" function.

The plug in spot receives a list of the ad hoc parameters for the batch job as name / value pairs. If the list includes parameters whose values are to be used in selecting records, your algorithm may be used to identify the relevant batch parameter passed as input and populate the field name and output bind variable appropriately.

The product provides a base algorithm type for this plug-in spot that simply defines a parameter for the SQL. It also includes parameters for the strategy and the key field name.


Copyright © 2026, Oracle and/or its affiliates.

This algorithm type may be used by any custom batch process where the SQL does not rely on any special bind variables that must be determined. Simply create an algorithm for the algorithm type and provide the appropriate SQL. Refer to the algorithm type Select Records by Predefined Query (F1-PDB-SR) for more information.

### Process Records Algorithm

The other important algorithm to design when implementing a plug-in driven batch process is the Process Record algorithm, plugged in on the batch control page. This algorithm is called for each record selected for the process. It receives all the information that was selected from the Select Records plug-in.

For the ad hoc processing batch process, algorithms plugged into this spot are responsible for doing the work for each record based on the desired logic.

For the extract batch process, algorithms plugged into this spot are responsible for returning the data that should be written to the file. Many of the parameters for this plug-in spot are only applicable to extract processes. Refer to Extract Record Processing for more information on the functionality specific to extract processes.

Also note that algorithms for this plug-in spot will be passed two Booleans, isFirst and isLast, to indicate if the current work unit is the first and/or last for that thread. This allows for the plug-in to do additional work if needed. Note that the isFirst indication is available for both types of batch processes, ad hoc and extract. However, the isLast indication is only applicable for the file extract batch. For the ad hoc batch process this value will always be set to **false**. Extracts will always execute in a single database transaction. In a single transaction run, any error causes the run to be aborted so that it restarts from the beginning when resubmitted. This is done to avoid partial files being written along with inaccurate setting of the isLast element.

### Parameter Validation Algorithm

Plug-in driven batch processes may require batch parameters specific to that process. The Parameter Validation algorithm is responsible for checking the parameters at run time and issuing errors if any values are invalid. The algorithm is invoked prior to the Select Records and Process Records algorithms. It is plugged in on the batch control portal. If multiple algorithms are configured, the one with the highest sequence is used. Algorithms for this spot should use standard error message reporting.

### Configuring a New Process

The following points summarize the steps needed to implement a new background process that acts on records in the system using plug-ins for the specific functionality:

- Verify the SQL that the background process should execute. Keep in mind that all the data selected in the SQL is available to pass to the plug-in that processes the records. If the performance of the background process is important, be sure to consult with a DBA when designing the SQL.
  - If the SQL does not require any custom variables to substitute at runtime, create an algorithm for the base algorithm type F1-PDB-SR and configure the SQL. In addition, configure the strategy and the primary key name (for the **Thread Level SQL Select** strategy).
  - If the SQL requires custom variables, a new plug-in script must be designed and coded to populate the variable names and values using the algorithm entity **Batch Control - Select Records**. Besides defining the variables, the algorithm must also indicate the strategy and the primary key name (for the **Thread Level SQL Select** strategy). Define the algorithm type for the newly created script. The first parameter of the algorithm type must be the SQL as illustrated in the base algorithm type. Note that


Copyright © 2026, Oracle and/or its affiliates.

the other parameters are available for use by this algorithm type if needed. Define the algorithm, populating the SQL as appropriate (using the custom variables).

- Design the logic required to process each record. If your edge product does not deliver a suitable plug-in, create a plug-in script where the algorithm entity is **Batch Control - Process Record**. Note that the plug-in receives all the information selected in the SQL defined in the Select Records plug-in. For an ad hoc process, the algorithm should perform whatever process is required based on the business use case. Note that the background process is responsible for committing the records.

  In either case, if a new processing script is required, define the algorithm type and algorithm for the newly created script.
- If the process has special parameters, create a script to determine if the parameters passed at run time are valid. Define the algorithm type and algorithm for the newly created script.
- Create a batch control by duplicating the base template `F1-PDBG`. Plug in the algorithms created above and configure the parameters as appropriate. Note that you may configure custom ad hoc parameters on the batch control if required. Both base and custom batch parameter values are available to the Select Records and Process Records plug in algorithms.

## Extract Record Processing

A process that selects records and produces output for each record to write to a file to integrate with external systems is called an Extract Process. The base batch control Plug-in Driven Extract Template (`F1-PDBEX`) may be used as a template.

> **NOT_SUPPORTED**
>
> Please read the topic Ad hoc Processes. Its information about the Select Records plug-in and some of the information about the Process Record plug-in apply to extract processes. This topic focuses on aspects that are unique to extract processes.

The system provided process includes parameters to configure the file path and file name for the created file. The file path supports all the functionality described in the topic Referencing URIs. The file name supports system substitution variables, such as run number, thread number, user and date / time. The process also supports compression if an appropriate extension is used. Refer to the File Name parameter description in the batch control for more information. Also see sections below for additional information for XML or JSON ouput formats.

> **Note**
>
> Refer to Flexible File Name / Writing Multiple Files for information about support for substituting a business value into the file name.

For extract processes, the Process Record algorithm is responsible for returning the data that should be written to the file in one or more XML instances along with the schema name(s) that describes the XML instance(s). The program will write the data to the file as per the format indicated in the File Format batch parameter.

By default the service uses the OUAF format for date and time. To override this and use XSD format, configure the Date Time Format batch parameter to 'XSD''.


Copyright © 2026, Oracle and/or its affiliates.

If an existing schema satisfies the output requirements, it may be used. Otherwise, you should define a data area to indicate the output format of the records as appropriate. Note that you should define an appropriate Field for each element in the schema. This is needed for the fixed position and for JSON / JSON lines formats, but is good practice for any output format. Refer to `F1-PDBGenProcExtractRecord` for an example of a data area with fields defined for each schema element.

There are two options for designing and configuring the Process Record algorithm:

- Create a specific **Process Record** algorithm type that encapsulates the extract logic. The product provides a base algorithm type that illustrates the basic technique to follow. Refer to the algorithm type General Process - Sample Process Record Extract (`F1-GENPROCEX`) for more information.
- Create a file integration type that defines the records to be included in the extract and configure a plug-in driven batch control that references that integration type. Refer to Extract Using File Integration for more information about implementing this type of functionality.

Your specific edge product may provide other Process Record algorithm types out of the box. Use the Algorithm Entity portal to find examples.

#### XML Format - Additional Considerations

When using XML format for the output, the batch process includes a parameter to define the Grouping Name. This is the root node to use to surround the whole output file. Refer to the batch parameter description for more information. When requesting a Multi-threaded Extract and using the concatenation option, the Grouping Name surrounds the full content of the concatenated output.

Each call to the Process Record plug-in returns one or more schemas with content. The content is written within a node using the schema name.

Your Process Record plug-in may return multiple schemas with information that is all part of the same overall record. For example, the output could include account information and related service agreement details:

```
<root>
...
<CM-AccountRecord>
<account type="group"
    <accountId>1234567890</accountId>
    ...
</account>
</CM-AccountRecord>
<CM-SAInfo>
<sa type="group"
    <saId>123457665</saId>
    ...
</sa>
</CM-SAInfo>
<CM-SAInfo>
<sa type="group"
    <saId>1234588913</saId>
    ...
</sa>
</CM-SAInfo>
```


Copyright © 2026, Oracle and/or its affiliates.

```
...
</root>
```

In this case, it may be desired to wrap all that information in a grouping XML tag so that all the information for one record is grouped together. Because the Process Record plug-in could be returning header or footer records that are not part of a given selected record's extract information, the batch process doesn't know what schemas returned by the Process Record plug-in belong together logically. The output parameter 'record XML node' in the schema collection should be used to indicate the outer XML node to use to group related information together. In this example, one call to the Process Record plug-in returns results with the node 'record' populated for the account and service agreement schemas:

```
<SchemaInstance>
<recordXMLNode>record</recordXMLNode>
<schemaName>CM-AccountRecord</schemaName>
<schemaType>F1DA</schemaType>
<data><account><accountId>1234567890</accountId>...</data>
</SchemaInstance>
<SchemaInstance>
<recordXMLNode>record</recordXMLNode>
<schemaName>CM-SAInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><sa><saId>123457665</saId>... </data>
</SchemaInstance>
<SchemaInstance>
<recordXMLNode>record</recordXMLNode>
<schemaName>CM-SAInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><sa><saId>1234588913</saId>... </data>
</SchemaInstance>
```

All schemas returned from a single call to the Process Record plug-in that have the same record XML node will be grouped in the written output within that XML tag. This snippet shows the results of two calls to the Process Record plug-in.

```
<root>
<record><CM-AccountRecord>
<account>
    <accountId>1234567890</accountId>
    ...
</account>
</CM-AccountRecord>
<CM-SAInfo>
<sa>
    <saId>123457665</saId>
    ...
</sa>
</CM-SAInfo>
<CM-SAInfo>
<sa>
    <saId>1234588913</saId>
    ...
</sa>
</CM-SAInfo>
</record>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<record>
<CM-AccountRecord>
<account>
    <accountId>987654320</accountId>
    ...
</account>
</CM-AccountRecord>
<CM-SAInfo>
<sa>
    ...
</sa>
</CM-SAInfo>
</record>
</root>
```

### JSON / JSON Lines Format - Additional Considerations

The extract process supports two types of JSON output.

- **JSON Document**. This format produces a valid JSON document. This output format supports returning a single JSON object with the records defined as an array or as an array of records.
- **JSON Lines**. The format produces a series of JSON objects. Each output record is a separate JSON object.

> **Note**
>
> These output formats do not support including a header or footer as part of the extract.

> **Note**
>
> The recommendation is to return only one schema instance for each call to the Process Record plug-in spot. This is because the service that converts from XML to JSON must have a schema name that describes the payload. If multiple schema instances are returned, it treats each returned instance independently and doesn't support nesting or other relationships between the data. For example, if the output returns one instance with Account data and then 3 instances with Service Agreement (SA) data (related to the account), in delimited, fixed position and XML, those four instances can be written one after the other. However, in JSON, the data for the 3 SA instances should be returned as an array within the Account object. In order to support this, the Process Record plug-in should have a governing schema that has the Account and the SA list all defined within it.

### JSON Document - Single JSON Object

If you provide a Grouping Name in the batch parameter, the program will produce a JSON object which contains an array of the output records using the grouping name as the array name.


Copyright © 2026, Oracle and/or its affiliates.

For example, imagine the Grouping Name is 'records' and the output from one call to Process Records is this:

```
<SchemaInstance>
<schemaName>CM-FileInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><files><fileName>file1.txt</fileName><externalReferenceId>1234</
externalReferenceId></files></data>
</SchemaInstance>
```

Now imagine three calls to the Process Record plug-in with separate results. In this case the 'files' node is the group node for the content of each call. This is mapped to the 'records' array name.

```
{
 "records": [
  {"fileName": "file1.txt", "externalReferenceId": "1234"},
  {"fileName": "file2.txt", "externalReferenceId": "3456"},
  {"fileName": "file3.txt", "externalReferenceId": "5678"}
 ]
}
```

The following are points related to how the schema returned by Process Records may be defined and how the JSON output format handles the use case. In all cases, the assumption is that the batch parameter Grouping Name is defined as 'records'. And in each case, the Resulting File Output imagines three calls to Process Records.

**Table JSON Output**

| Use Case | Schema Definition | Resulting File Output |
|---|---|---|
| If the output schema contains a single group, output returned by the plug-in is grouped by the a group node. This node does not have importance when writing the record to file and the group node is removed. The record is written directly inside the JSON array. | <schema><br>  <group1 type="group"><br>  <fileName dataType="string"/><br><externalReferenceId dataType="number"/><br>  </group1><br></schema> | The 'group1' node is dropped and its contents are mapped to the 'records' array.<br><br>{<br>"records": [<br>{"fileName":"file1.txt",<br>"externalReferenceId":"1<br>,234"},<br>{"fileName":"file2.txt",<br>"externalReferenceId":"1<br>,234"},<br>{"fileName":"file3.txt",<br>"externalReferenceId":"1<br>,234"}<br>]<br>} |


Copyright © 2026, Oracle and/or its affiliates.

**Table (Cont.) JSON Output**

| Use Case | Schema Definition | Resulting File Output |
|---|---|---|
| If there is an outer group node in the schema which contains nested group(s) within it, the program skips the topmost group node. | <schema><br>  <group1 type="group"><br>  <group2 type="group"><br>  <fileName dataType="string"/><br><br><externalReferenceId dataType="number"/><br>  </group2><br>  </group1><br></schema> | {<br>"records": [<br>{"group2":<br>{"fileName":"file1.txt",<br>"externalReferenceId":"1<br>,234"}},<br>{"group2":<br>{"fileName":"file2.txt",<br>"externalReferenceId":"1<br>,234"}},<br>{"group2":<br>{"fileName":"file3.txt",<br>"externalReferenceId":"1<br>,234"}}<br>]<br>} |
| If schema contains more than one group or a group along with other types like list, field or raw then it does not skip any group node name. | <schema><br>  <group1 type="group"><br>  <fileName dataType="string"/><br><br><externalReferenceId dataType="number"/><br>  </group1><br>  <list1 type="list"><br>  <fileDetails dataType="string"/><br>  </list1><br>  <fileSize dataType="number"/><br></schema> | All nodes are included in each entry in the array.<br><br>{<br>"records": [<br>{"group1":<br>{"fileName":"file1.txt",<br>"externalReferenceId":"1<br>,234"},"list1":<br>[{"fileDetails":"Test<br>file"}],"fileSize":"1,02<br>4"},<br>{"group1":<br>{"fileName":"file2.txt",<br>"externalReferenceId":"1<br>,234"},"list1":<br>[{"fileDetails":"Test<br>file"}],"fileSize":"1,02<br>4"},<br>{"group1":<br>{"fileName":"file3.txt",<br>"externalReferenceId":"1<br>,234"},"list1":<br>[{"fileDetails":"Test<br>file"}],"fileSize":"1,02<br>4"}<br>]<br>} |

**JSON Document - Array of Output**


Copyright © 2026, Oracle and/or its affiliates.

If you choose the JSON document output format and populate 'suppress' for the Grouping Name, the program will produce an array instead of an object.

To illustrate, we assume the same definitions used in the above Single JSON object format examples. The following is the output of one call to the Process Records plug-in.

```
<SchemaInstance>
<schemaName>CM-FileInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><files><fileName>file1.txt</fileName><externalReferenceId>1234</
externalReferenceId></files></data>
</SchemaInstance>
```

Now imagine three calls to the Process Record plug-in with separate results. In this case the 'files' node is the group node for the content of each call. This node is dropped and the output from each call is an object within the array.

```
[
  {"fileName": "file1.txt", "externalReferenceId": "1234"},
  {"fileName": "file2.txt", "externalReferenceId": "3456"},
  {"fileName": "file3.txt", "externalReferenceId": "5678"}
 ]
```

The same considerations in the JSON Output table above apply to this configuration.

**JSON Lines**

For the JSON Lines output format, the Grouping Name is not applicable (and ignored). Each call to the Process Record plug-in spot is written as an object. All the considerations in the JSON Output table above apply to this configuration.

To illustration, here is the output for a single call to the Process Record plug-in spot.

```
<SchemaInstance>
<schemaName>CM-FileInfo</schemaName>
<schemaType>F1DA</schemaType>
<data><files><fileName>file1.txt</fileName><externalReferenceId>1234</
externalReferenceId></files></data>
</SchemaInstance>
```

Then this is an example after three calls to Process Record. The output is written out as objects with no commas.

```
{"fileName": "file1.txt", "externalReferenceId": "1234"}
{"fileName": "file2.txt", "externalReferenceId": "3456"}
{"fileName": "file3.txt", "externalReferenceId": "5678"}
```

**Multi-threaded Extract**

When extracting data, it's possible that the volume of data warrants running the job multi-threaded. In this case, a separate file is produced with the thread number included in the file name. You may proactively include the thread number as a substitution variable when indicating the file name. If not, the system appends the thread number.


Copyright © 2026, Oracle and/or its affiliates.

As a convenience, the system supports concatenating the extract files produced by the various threads at the end. The system does this by finding files whose file names match except for the thread number. As such, the file name may not contain date or time if the concatenation parameter is true. Please note the following with respect to the concatenation feature:

- The content of each file is concatenated together "as is". If the individual files have header, footer or summary information, there is no logic to consolidate that information in the concatenated file. If you want a single header, footer or summary, you must run the extract single threaded.
- Concatenation is not supported if the file name indicates the zip compression should be used.
- The individual thread files are retained.
- If the file format is XML, the files written for the individual threads will be written with an extension of ".tmp". The final concatenated file is surrounded by the Grouping Name defined on the batch job as a parameter (if not suppressed).
- If the file format is JSON Document, the files written for the individual threads will be written with an extension of ".tmp". The final concatenated file is treated as an object with an array defined by the Grouping Name, or treated as a single array (if the grouping name is configured as "suppress").

### Skipping Records

By default, the extract process expects one or more schemas to be returned by each call to the Process Record plug-in. (Ideally when designing any process, the Select Records algorithm only selects records that should be processed and the Process Record algorithm does not need to confirm whether the record should be part of the processing or not.) If however there is a need to check a condition in the Process Record plug-in and it is found that no data needs to be extracted for this record, the process record algorithm should return the 'is skipped' output parameter to **true** so that the program doesn't write an empty row.

Note that the base process records algorithm for file integration types `(F1-FILEX-PR)` will set the ‘is skipped’ parameter to **true** if the file integration record extract record algorithm returns no data.

### Configuring a New Process

The points documented in the topic related to configuring a new process with respect to the Select Records algorithm apply to extract processes.

For the Process Records algorithm, it is responsible for returning one or more schema instances populated with information that should be written to the file. If your edge product does not deliver a suitable plug-in, create a plug-in script where the algorithm entity is **Batch Control - Process Record**. Note that the plug-in receives all the information selected in the SQL defined in the Select Records plug-in.

Refer to Configuring an Extract Process Using File Integration Type for information on designing extracts using this configuration.

If a new processing script is required, define the algorithm type and algorithm for the newly created script. Create a batch control and refer to the batch template Plug-in Driven Extract (`F1-PDBEX`). Plug in the algorithms created above and configure the parameters as appropriate. Note that you may configure custom ad hoc parameters on the batch control if required. Both base and custom batch parameter values are available to the Select Records and Process Records plug in algorithms.

- Flexible File Name / Writing Multiple Files


Copyright © 2026, Oracle and/or its affiliates.

- Extract Using File Integration

## Flexible File Name / Writing Multiple Files

By default, the data is written to the File Name defined in the batch parameter. The file name supports several dynamic substitution variables as defined in the detailed description for the batch parameter in the Plug-in Driven Extract Template batch control (F1-PDBEX). Also refer to Extract Record Processing for information about the file name and multi-threaded processes.

The system also supports more advanced file creation for a single batch run. Consider the following use cases

- An extract is for data associated with multiple service providers where each service provider should receive its own file. In this use case, the unit of work is the service provider so a file is produced for each work unit.
- An implementation operates with multiple jurisdictions represented by CIS Division. A certain extract should result in a file per division so that each division only gets its relevant data.
- An extract of Person information should separate individual person data from business person data. In addition, for each person, the contact information should result in a separate file. In this use case, one unit of work includes data to be written to more than one file.

To support the above use cases, the Process Record algorithm supports returning the File Name in the output:

- The file name is included in the output Schema Instance group. You would populate this if you want the data for a specific Schema entry to go to this file name. In the person extract example above, the schema that includes the contact information would indicate the specific contact file.
- The file name is an output field outside the schema list. You would populate this if you don't have different files for different schemas but you still want to override the file name defined in the batch parameter. The service provider and division examples above would populate this parameter with the desired file name. The person extract example could populate this field for all the non-contact information.

The batch process checks the output after each call to the Process Records algorithm. For each schema, if there is a specific File Name associated with the schema, the data for that schema is written to that file. Otherwise, if the output file name is provided, the data is written to that file. Otherwise, the File Name in the batch parameter is used.

> **Note**
>
> **Substitution Variables.** After the call to Process Records, if there are any system variables in the file name, they are resolved at this time.

### Managing File Names

This type of requirements means that the Process Record algorithm needs to provide a file name that has some business value to distinguish the individual files. For example, the service provider or the division or the person type value must be part of the file name. However, the recommendation is not to have the file name entirely determined by the algorithm. Rather the suggestion is that you still allow the user submitting the batch job to provide a desired file name as a batch parameter. That way, it can include any of the supported substitution variables for system information (such as batch run number, or date /time stamp) that ensure


Copyright © 2026, Oracle and/or its affiliates.

your files are unique for a given batch run. The algorithm can designate a substitution variable that it looks for to replace with the business value.

For example, using the files per division use case, imagine the user indicates the file name EXTRACT_FILE_<DIV>_{BN}_{RN}_{TN}.csv. The process record algorithm receives this parameter. It knows to look for "<DIV>" and replace it with the actual division. Imagine the division code is A100, it would return the file name EXTRACT_FILE_A100_{BN}_{RN}_{TN}.csv in its output, along with the data to write to that file for this unit of work.

For the person information use case above, where two files are written, the suggestion is to use the File Name parameter for one of the files, for example the person information file. An additional parameter should be used to define the second file, the contact information file. So the users submitting the batch job would define the following parameters:

| Parameter Name | Description | Value | Comments |
|---|---|---|---|
| **fileName** | File Name | persons_<personType>_{BN}.txt | This is the file name parameter provided with the extract. |
| **contactFileName** | Contact File Name | contacts_{BN}.txt | This is a special parameter defined by the CM extract batch control. |

The Process Record algorithm would look for the File Name parameter provided as input, find the text "<personType>" and replace it with the person type value and return that file name in the output. When building the list of schemas to return in the output, the one that includes the person contact information would include the file name found in the Contact File Name parameter provided as input.

> **Note**
>
> If an extract uses file integration types, the file integration record's extract process algorithm would have the above responsibility for indicating the file name and pass it back to the Process Record algorithm.

### Multiple Threads

This functionality works as expected with multiple threads. Based on the use case and the implementation of the Select Records algorithm, each thread could be producing files for the same business value. Let's use the division use case. Imagine an implementation has three divisions: A100, B200, C300. Now imagine that the extract is for financial information for an account. The select records algorithm should be threading by account id so that the data is evenly distributed across the multiple threads. But within each thread, accounts could be in any of the 3 implementation's divisions. So each thread could produce 3 files for each of the 3 divisions.

As described in Multi-threaded Extract, the system supports an option to concatenate, allowing you to consolidate the files such that you have 3 complete files for the 3 divisions. The same features and limitations described in that topic apply for this use case.

### Closing Files

An important consideration when using this functionality is that attention should be paid to how many files are open during the batch process. Each open file takes some memory to manage it. As such, the product has imposed a limit of only 10 open files (per thread) during the run of


Copyright © 2026, Oracle and/or its affiliates.

the batch job. When designing a batch to write to multiple files, the product strongly encourages you to design the process so that whenever possible, all the data for one file can be written and that file can be closed before going on to the next file.

Let's look at our examples.

- For the service provider example, the unit of work for the batch process is service provider. In this case, after each call to process records, the data can be written to the file and then the file can be closed.
- For the division example, the data selected should be threaded by the proper unit of work related to the purpose of the extract, such as Account or Asset. But the data should be ordered by division so that all the data for each division is processed before the next division's data. Each division's file can be closed once the next division's data starts getting processed.
- For the person extract example, the contact file is used for all the data so that file needs to remain open throughout the duration of the job. However, the person information gets segrated by person type. The Select Records could order the data by person type so that all the data for one type could be written and then all the data for the other type could be written. However, given that for any run no more than 3 files will ever be open, this use case is one where there is no need to actively manage closing files during the process.

How will the batch program know when to close a file? Since the batch program is agnostic and does not know what type of data is being processed nor when files should be closed, the Process Record algorithm should provide this information.

> **Note**
>
> The batch process will issue an error if Process Records indicates that the file to write to is one that was already closed. Reopening closed files is not allowed. The design of the process needs to be reviewed if this error is received.

To manage this, the batch process provides the Process Record algorithm with the list of open files (with the system substitution variables unresolved) and a Boolean to mark whether the file should be closed.

> **Note**
>
> If an extract uses file integration types, the file integration record's extract process algorithm receives this list of open files and would have the responsibility described here to manage closing files.

To explain further, we'll use our examples.

**File Per Division Example**

This use case is that there are 4 divisions in the implementation: NORTH, EAST, SOUTH, WEST and that the data (account data) should be grouped in separate files by division.

The recommendation is that the Select Records algorithm selects the data ordered by Division.

The batch process passes to the Process Record algorithm the list of open files (there should be only be zero or one). In this case, the only indication that the data for one division is finished is if the Process Record algorithm sees that its data is for one division and the open file listed is for a different division. Each call should prepare its file name. Then it should see if its file


Copyright © 2026, Oracle and/or its affiliates.

name is in the list of open files. If not, it should indicate that the other open file should be closed. (Note that it doesn't need to add itself to the 'open list'. The batch program is only looking for entries marked as closed. It will ignore entries that are not marked as closed).

Assume the file name in the batch parameter is myFile_<DIV>_{BN}.txt

The data is sorted so that the accounts in the EAST division are processed first. Let's imagine this is a work unit that is also in the EAST division. It receives the file name and the list of open files.

```
...
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>myFile_<DIV>_{BN}.txt</value>
  </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <closeFile></closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

The algorithm substitutes its division into the <DIV> portion of the file and populates that in the output. It then compares its file name with the current list of open files. It finds its file so it doesn't need to do anything else.

```
<hard>
...
<fileName>myFile_EAST_{BN}.txt</fileName>
...
<openFiles>
  <OpenFile>
    <closeFile></closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

If we then imagine a call to the process records algorithm where the division is NORTH and it's the first entry for that division, here is what it receives:

```
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>myFile_<DIV>_{BN}.txt</value>
```


Copyright © 2026, Oracle and/or its affiliates.

```
  </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <closeFile>true</closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

The algorithm substitutes its division into the <DIV> portion of the file and populates that in the output. It then compares its file name with the current list of open files and doesn't finds its file name. So it indicates that the existing open file should be closed.

```
<hard>
...
<fileName>myFile_NORTH_{BN}.txt</fileName>
...
<openFiles>
  <OpenFile>
    <closeFile>true</closeFile>
    <fileName>myFile_EAST_{BN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

**File Per Service Provider Example**

In this use case, the unit of work is service provider and each service provider should have its own file. In this case, as the Process Record algorithm is preparing the data and the file name unique to its service provider, it can also indicate that the file can be closed. There is no need to wait for the next call to Process Record to indicate that the previous file can be closed. Let's assume the file name is MYFILE_<serviceProvder>_{BN}_{TN}.txt. Each call to Process Records will pass the batch parm file name and no open files.

```
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>MYFILE_<serviceProvider>_{BC}_{TN}.txt</value>
  </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <fileName></fileName>
    <closeFile></closeFile>
  </OpenFile>
</openFiles>
```


Copyright © 2026, Oracle and/or its affiliates.

```
...
</hard>
```

The algorithm is processing data for service provider A100. It returns the following information related to the file name, namely the single File Name in the output related to this service provider and adds its file to the open file list and indicates that it should be closed. (Note that file-per-work-unit is the only known use case where the algorithm will know that it is the last entry for a file and to close itself.)

```
<hard>
...
<fileName>MYFILE_A100_{BC}_{TN}.txt</fileName>
...
<openFiles>
  <OpenFile>
    <closeFile>true</closeFile>
    <fileName>MYFILE_A100_{BC}_{TN}.txt</fileName>
  </OpenFile>
</openFiles>
...
</hard>
```

**Multiple Open Files Example**

This example assumes that the file names are defined as per the Managing File Names section above. We'll assume that the algorithm designer chose to not actively close the person information file since there are never more than three files open. Similarly, we'll assume that they chose not to order the data by person type. Here is what is passed in in the middle of the run:

```
<hard>
<batchParms>
  <BatchParm>
    <name>fileName</name>
    <value>persons_<personType>_{BN}.txt</value>
  </BatchParm>
  <BatchParm>
    <name>contactFileName</name>
    <value>contacts_{BN}.txt</value>
   </BatchParm>
  ...
</batchParms>
<openFiles>
  <OpenFile>
    <fileName>persons_P_{BN}.txt</fileName>
    <closeFile></closeFile>
  </OpenFile>
   <OpenFile>
    <fileName>persons_B_{BN}.txt</fileName>
    <closeFile></closeFile>
  </OpenFile>
  <OpenFile>
    <fileName>contacts_{BN}.txt</fileName>
    <closeFile></closeFile>
  </OpenFile>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</openFiles>
...
</hard>
```

An instance of one call to Process Records algorithm that has details for an individual ("P" person type).

```
<fileName>persons_P_{BN}.txt</fileName>
<fileOutput>
  <listValue>
    <SchemaInstance>
      <recordXMLNode>person</recordXMLNode>
      <schemaName>CM-PersonDetail</schemaName>
      <schemaType>F1SS</schemaType>
      <fileName></fileName>
      <data>LotsOfData</data>
    </SchemaInstance>
  </listValue>
  <listValue>
    <SchemaInstance>
      <recordXMLNode>person</recordXMLNode>
      <schemaName>CM-AddressDetail</schemaName>
      <schemaType>F1SS</schemaType>
      <fileName></fileName>
      <data>LotsOfData</data>
    </SchemaInstance>
  </listValue>
  <listValue>
    <SchemaInstance>
      <recordXMLNode></recordXMLNode>
      <schemaName>CM-ContactDetail</schemaName>
      <schemaType>F1SS</schemaType>
      <fileName>contacts_{BN}.txt</fileName>
      <data>LotsOfData</data>
    </SchemaInstance>
  </listValue>
</fileOutput>
```

This XML example illustrates the following points:

- There is an override file name in the fileName element.
- The first two schemas don't have any value in the fileOutput\listValue\SchemaInstance\fileName so they should be written to the file in fileName. Note that these two schemas also include a Record XML Node. That will ensure that if the output is XML format, the data from both of those schemas will be wrapped by the 'person' XML Node. This is not functionality related to this epic. It's just trying to illustrate multiple features.
- The third schema has an entry in fileOutput\listValue\SchemaInstance\fileName . So that data is written to a separate file.


Copyright © 2026, Oracle and/or its affiliates.

# Extract Using File Integration

The product provides separate configuration that may be used to define how to extract the data and how to format it. This configuration provides more metadata related to defining how to get the data to extract and how to format the data. You may have many use cases where using this configuration rather than having all the logic directly in the Process Records algorithm is beneficial.

The configuration can support simple extracts, containing a single record type and possibly a header or footer record. It also supports more complex integrations, involving multiple record types and parent/child relationships between the records.

### File Integration Type Configuration

The framework supplies two objects that work together to define the processing needed to create a specific type of extract:

- The file integration type which describes and classifies the extract configuration.
- The file integration record where the sequence, format and logic to extract the data required for a given record type are captured.

### File Integration Type

The file integration type defines the integration classification. Currently, only **Extract** is supported. In addition, the file integration type can be assigned to a category. The category may be used to identify extracts with a common purpose or target such as collection agency referrals. Your edge product may deliver a number of categories or you may define your own.

The system provides a business object for extract file integration types (**`F1-ExtractFileIntegrationType`**) that is expected to be used for most use cases

### File Integration Record

The file integration record is the object that is responsible for defining a record's format and capturing the record details. A separate integration record must be defined for each record type included in the extract.

The record sequence defines the order that records are processed by the system. The parent sequence defines the record type of which this record is a child. Child records may also be referenced as a parent of lower level child records. The base Process Records algorithm provided by the framework processes all child records of a record type before processing the next record type in sequence.

For extracts that are XML format, the system supports creating a grouping node surrounding all the output that belongs to the same entity. File integration records define a record XML node that will be used as the grouping node. Each record type in the list that belongs within the same group would indicate the same value in the record XML node. Refer to theXML Format - Additional Considerations section for more details.

> **Note**
>
> The system does not support nested record XML nodes. All child records of a record type must define the same record XML node as the parent.

Each file integration record references a data area that defines the extract record’s format, allowing the data to be extended. The **Extract Records** algorithms are responsible for


Copyright © 2026, Oracle and/or its affiliates.

populating the data area. The file integration record supports multiple extract processing algorithms so that the custom information can also be populated.

The extract record plug-in also supports the functionality described in the Flexible File Name / Writing Multiple Files topic.

The system provides a business object for extract file integration records (**`F1-ExtractFileIntegrationRec`**) that is expected to be used for most extract record use cases.

#### Configuring an Extract Process Using File Integration Type

The following points summarize the additional steps needed to implement a new extract background process that uses a file integration type to define the specific functionality:

- Choose or create data area objects to define the extract record formats.
- Define the logic required to populate each extract record type. Your edge product may deliver suitable plug-ins or you may need to create a plug-in script where the algorithm entity is **File Integration Record - Extract Record**. Note that the Extract Record plug-in receives most of the information defined in the Batch Control - Process Records plug-in.
- Define the algorithm type and algorithm for each newly created script.
- Choose or create a category for the file integration type using lookup `FILE_INT_OBJ_FLG`.
- Create the file integration type and configure the file integration records using the defined data areas and algorithms.
- Create a batch control by duplicating the appropriate base template for an extract background process. Include the File Integration Type parameter (fileIntegrationType) on the batch control and reference the appropriate file integration type.
- Plug in a Process Records algorithm that reads the file integration type configuration and processes the record types. The system provides a base file integration record extraction algorithm (`F1-FILEX-PR`) that is expected to be used for most file integration plug-in extract processes.

#### Extending a File Integration Extract Process

Your edge product may deliver file integration types and batch controls designed for standard extract processes. However, extending the delivered extracts to add custom data is common practice for most implementations. You can customize the output from a delivered extract by:

- Identifying the record types that need additional details
- Creating a data area that defines the additional details and using that to extend the data area configured on the record type
- Creating a File Integration Record — Extract Record plug-in script, algorithm type and algorithm that populates the extra details
- Adding the algorithm to the integration record's extract algorithms collection

In some cases, you may wish to extend the extract by adding additional record types. To do this, add your custom file integration records to the file integration type, following the steps described in Configuring an Extract Process Using File Integration Type.

## Uploading Records

The base product provides a background process to upload data from a file. The batch control Plug-in Driven File Upload Template (`F1-PDUPL`) may be used as a template. The process


Copyright © 2026, Oracle and/or its affiliates.

includes parameters to configure the file path and file name for file to upload along with other parameters to control how to handle missing files and how to rename the file once processed. The process supports uploading multiple files and may be run multi-threaded to process multiple files. Refer to the description of the batch control and its parameters for more information.

The following sections describe concepts and features related to this process.

### A Step in the Overall Upload Process

This step in the upload of data is only one part of a typical upload end-to-end process. This step is sometimes referred to in the product as "Process X". The goal of this step is to get records from a file into database records, with minimal validation and processing. The data should be stored in records that are then processed by a second step (often referred to in the product as the "upload" step, for example "Payment Upload"). The second step, independent of the plug-in driven batch process described here, is responsible for validating the data and should be able to be threaded by individual records and have proper error handling for individual records.

Depending on the type of data being uploaded, the product may already supply appropriate tables that the plug-in driven upload process may populate. These could be staging tables, such as payment upload staging. Or they may be records with business objects that have a lifecycle designed to handle uploaded data, for example Business Flag. In such cases, the product will typically supply out-of-the-box background processes to validate and further process the data and finalize the upload. If the data to upload does not already have a base provided staging table, be sure to work with your implementation team to identify an appropriate table to use for the plug-in driven batch upload. In addition, confirm the design for the second step that is responsible for the detailed validation and finalization of the data.

> **Note**
>
> Designing your upload process to directly update data, as opposed to using a staging table, increases the likelihood of data validation exceptions. Refer to the **Exception Handling** section for more information and guidelines to be considered when designing such upload process.

### The File Upload Algorithm

This background process requires an algorithm plugged into the plug-in spot **File Upload**. By default, this plug-in is called once for a given file but it can also be designed to process records in multiple unites of work.

The batch process opens the file and passes to the algorithm the file reader element. The algorithm is responsible for using provided APIs to read the content of the file and store the data in appropriate table(s). The algorithm can be designed to process all the records in the file as a single unit of work. Alternatively, it can be designed to process a smaller set of records and request to be called again to process the next set etc, until all records are processed.

The batch process commits the records uploaded by each call to the algorithm and keeps track of the position in the file of the last processed record. That position is provided to the algorithm with each subsequent call, allowing it to reposition the file to that record and continue processing from there. When all records in the file are processed the batch process closed the file and proceeds to the next file if any.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Plug-in scripts written to implement this type of algorithm must use the Groovy script engine version as the APIs are not accessible using the XPath scripting language. Refer to Additional Coding Options For Server Scripts for more information.

### Sample Algorithms

The product supplies sample algorithms to illustrate calling the supplied APIs for processing different types of source data: comma delimited, fixed position and XML. In every case, the sample data supported by the upload uses 'degree day' information to illustrate the process. The system provides sample target records (based on the Fact maintenance object) in order to illustrate the step to store records based on the input data. Note that only sample plug-in scripts have been provided. No algorithm, algorithm type or batch control is configured to use the sample plug-in scripts. To view the scripts, navigate to the Script page, search for a Script Type of **Plug-in Script** and Algorithm Entity of **Batch Control - File Upload** and look for the 'Sample' scripts.

Note that the sample plug-in scripts provided by the product are supplied to illustrate use of the provided APIs. They should not be considered a template for how to implement a plug-in script for a real use case. The following highlight some points to consider when designing a file upload algorithm:

- **Error handling / resolution.** The sample plug-in scripts do some basic error handling related to the data to illustrate error handling. However, any errors found in this step require processing of the whole file to stop. As such, this plug-in should only report errors that are not possible to fix, but where the whole file should be rejected. If there are errors that can be adjusted in the data, then the recommendation is to not check for those errors at this step. Rather, this plug-in should simply populate the appropriate staging tables and let the next step check for validity. As described above, the next step should include the ability to mark individual records in error, allowing for users to fix the data and retry.
- **Target tables.** The sample plug-in scripts use Fact as the target for the resulting insert statements. As mentioned above, the decision of where to store the uploaded data must be carefully considered. There may already be existing tables that are specific to a given use case. If the data being uploaded does not have existing tables to use, review the product to verify what existing tables may be useful, such as Inbound Sync Request or Service Task. Be sure that the tables chosen support error handling, either out-of-the box or via designing an appropriate business object with a lifecycle that supports an error status and the ability to resolve the error. Also note that the Sample Flat File Upload plug-in illustrates a header record / detail record scenario. In this case, the header record is linked to the child record via a CLOB element. This is not the recommended technique. In a real use case, the header record should be linked to the child record via a separate database column to allow for searching.

### Exception Handling

By default, any error that occurs while uploading a file causes the entire file to be skipped. The upload batch program managed an exception as follows:

- A rollback is issued to reverse updates since last commit.
- The error message is reported as part of the batch run.
- The file is skipped from further processing and its name remains unchanged.
- Since the default unit of work for an upload process is a file, and not an individual record within it, an error counts as one file unit in error towards the **Override maximum errors (maxErrors)** limit if specified as a batch parameter. If a limit was not specified or not


Copyright © 2026, Oracle and/or its affiliates.

reached, the batch run is considered completed regardless of how many files were successfully uploaded. If a limit has reached, the upload process is stopped and the batch run status is set to error.

If your upload process requires a more granular row level exception handling, your upload algorithm should be designed accordingly. With this approach, errors are fully trapped, managed and reported by your algorithm and never reach the upload batch program. In other words, to the batch process there were no errors.

Trapping errors by your custom upload algorithm is not trivial and should consider the following guidelines:

- Use the 'Savepoint Script Dispatcher' (F1-SavePointDispatcher) business service to perform all your updates along with a request to trap any errors. The service adheres to standard rollback and data integrity rules and as such is the recommended approach for trapping application errors.
- When the business service traps an error, your algorithm may report it as a batch run message or any other option that meets your business requirements.

> **Note**
>
> Do not report the number of errors as part of the algorithm's plug-in spot response 'unitsInError' element. Any value provided in this element counts as if the entire file is in error, i.e. increments the total number of errors by one towards the 'maxErrors' limit if specified.

- The upload batch process is not aware of errors trapped by your algorithm and will rename the file as successfully processed even if no record was successfully processed. It is the responsibility of your algorithm to determine when a file should be considered as not processed and issue an explicit error for the batch program to handle.

### Splitting Large Files Prior to Upload

A single file is processed by a single thread of this batch process. Splitting a large file into smaller files allows the smaller files to processed by multiple threads of the upload process.

The base product provides a background process to split large files prior to upload. The batch control File Split Template (`F1FSPLIT`) may be used as a template. The process supports splitting multiple files and may be run multi-threaded to process multiple files. The process can also be set up to submit the subsequent upload process once the files are split.

Note that not all file contents can be split using this batch process. Basic XML, json and flat file records separated by a new line may be supported.

Refer to the description of the batch control and its parameters for more information.

### Configuring a New Process

The following points summarize the steps needed to implement a new file upload background process:

- Verify the details of the data in the upload file and map the data to fields in one or more appropriate tables in the system.
- Design the logic required for reading the record details and identifying each record to properly create the insert statements for storing the data. The sample plug-in scripts provided by the product illustrate using the various APIs available for use. Create a plug-in


Copyright © 2026, Oracle and/or its affiliates.

script where the algorithm entity is **Batch Control - File Upload**. Create an appropriate algorithm type and algorithm for this plug-in script.

- Create a batch control by duplicating the base template. Plug in the algorithm created above and configure the parameters as appropriate. Note that you may configure custom ad hoc parameters on the batch control if required. Both base and custom batch parameter values are available to the File Upload algorithm.

## How to Re-extract Information

If you need to recreate the records associated with an historical execution of an extract process, you can - simply supply the desired batch number when you request the batch process.

## How to Submit Batch Jobs

Most batch jobs are submitted via a batch scheduler. For cloud customers, refer to the Batch Scheduler for Cloud Implementations topic in the Oracle Utilities Cloud Service Foundation documentation for more information. For on premises customers, refer to your third party scheduler documentation for more information.

Batch jobs may be configured as Timed, which means they will automatically be run based on the timer frequency.

You can manually submit your ad hoc background processes or submit a special run for one of your scheduled background processes using the online batch job submission portal.

The system provides a service script `F1SubmitJob` to create a batch job submission request. Notes about this service:

- No parameter defaulting occurs when submitting a record using this script. All the desired parameter values must be provided.
- If the requested batch control is defined as part of a pool of batch controls using the **Batch Group** option then the batch job request is created for a batch control within the pool that has a low number of in progress requests. This technique allows for a more balanced load of submission requests across pool-based batch controls.

The script is also exposed as a REST web service (using the inbound web service `F1-SubmitJob`) to support batch job submission requests from an external system.

## How to Track Batch Jobs

You can track batch jobs using the batch process pages, which show the execution status of batch processes. For a specified batch control id and run id, the Batch Run portal shows information about each thread, the run-instances of each thread, and any messages (informational, warnings, and errors) that might have occurred during the run.

> **NOT_SUPPORTED**
>
> For more information, refer to Tracking Batch Processes.


Copyright © 2026, Oracle and/or its affiliates.

# How to Restart Failed Jobs and Processes

Every process in the system can be easily restarted if it fails (after fixing the cause of the failure). All you have to do is resubmit the failed job; the system handles the restart.

# Assessing Level of Service

For some background processes, an implementation may wish to supply one or more algorithms that check some conditions to assess whether or not the process is performing as expected. An algorithm could be used to check the performance of the job to see if it is running as efficiently as expected. Or it could analyze the data processed by the background process to assess whether there may be some problem with the quality of the data.

The system provides a Level of Service plug-in spot on batch control to configure the appropriate algorithms for a given background process, if desired. Each algorithm is expected to return a value to indicate the 'level of service' determined along with a message indicating the reason for the value. The following Level of Service values are supported:

- **Normal**. Indicates that the algorithm did not detect any issues.
- **Warning**. Indicates that the algorithm found some issues that may or may not indicate a problem.
- **Error**. Indicates that the algorithm found some issues that should be investigated.
- **Disabled**. Indicates that the algorithm could not properly execute the level of service logic.

When multiple Level of Service algorithms are plugged in on the batch control, the overall Level of Service assessment for a batch control is populated as follows: if any algorithm returns Error, that value is returned; otherwise, if any return a Warning, that value is returned; otherwise Normal is returned. The results returned by each algorithm are also made available for review.

When viewing a batch control record, use the **Level of Service** action to display this information. The level of service is also part of the Health Check service.

# Batch Resource Groups

Batch resource groups may be configured to prioritize resource allocation between various batch processes. For example, export type of processes may have a lower resource allocation priority relative to billing related processes.

The **Batch Resource Configuration (F1-BatchResourceConfig)** extendable lookup associates certain batch processes with one of three priority resource groups: 10 (High), 20 (Medium), 30 (Low). No setting for a batch control means it received the default priority relative to other processes.

Resource group information may be leveraged as follows:

- Database resource management tools may be configured to allocate resources to batch processes by their association to resource groups.
- In a Cloud installation, thread allocation prioritization can be fine-tuned by associating batch processes to batch resource groups.

The following sections provide additional information on these features.


Copyright © 2026, Oracle and/or its affiliates.

### Batch Thread Prioritization

In a Cloud installation, batch processing functionality supports the capability of submitting batch jobs with high thread counts which leads to many threads of work being initially queued by the system. Batch resource groups are used to promote fair queuing of batch work such that batch controls associated with a higher priority resource group receive more service relative to those associated with lower priority resource groups.

### Database Resource Management Tools

Database resource management tools may be used to prioritize resource allocation between various batch processes. For example, export type of processes may have a lower resource allocation priority relative to billing related processes. These tools allocate resources to groups of consumers, not to an individual batch process. They use rules that are based on certain session variables, to associate a batch process with a resource group.

In a Cloud installation, if a batch control is associated with a resource group the latter is included by default as part of the MODULE database session variable as follows: '<batch control>,R=<resource group>'. This allows DBAs to potentially take advantage of this information for resource prioritization purposes. On premise, this feature needs to be explicitly opted in using the **Expose Batch Resource Group** option in the **General System Configuration** feature configuration.

With this configuration enabled, DBAs can set up rules that parse the MODULE variable to identify the resource group for a batch process and apply prioritization rules accordingly. Note that most batch controls may not be associated with a resource groups so default allocation rules should be considered.

**Note.** This configuration does not include nor enforce the database resource management configuration. It only allows for such tool to be leveraged by DBAs as needed.

## Batch Control Options

A batch control supports defining options. An option allows the product or your implementation to mark a batch control with additional information to be used for special processing or reporting.

**Note. You can add new option types.** Your implementation may want to add additional batch control option types. To do that, add your new values to the customizable lookup field **BATCH_CTRL_OPT_FLG** .

# Defining Batch Controls

The system is delivered with all necessary batch controls. Implementations may define default values for parameters. In addition, implementations may define their own background processes.

To view background processes, open **Admin**, and then **System**, and then **Batch Control.**

Use the **Batch Control Query** portal to search for an existing batch control. Once a batch control is selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to Background Processing Concepts and Online Batch Submission for more information.


Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! If you introduce a new batch process, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Batch Control.** This zone provides information about the selected batch control.

The following zone may appear as part of the portal's **Algorithms** tab page

- **Algorithms.** This zone is used to maintain the algorithms that control important functions for instances of this batch control. Refer to the table that follows for a description of all possible system events. If the algorithm is implemented as a script, a link to the **Script** is provided. Refer to Plug-in Scripts for more information.

The following zone may appear as part of the portal's **Options** tab page

- **Options.** This zone is used to maintain various configuration options associated with the current batch control.

> **ⓘ Note**
>
> **You can add new option types.** Your implementation may want to add additional batch control option types. To do that, add your new values to the customizable lookup field `BATCH_CTRL_OPT_FLG`.

The following zone may appear as part of the portal's **References** tab page

- **Batch Control References.** This zone list various configuration entities referencing the current batch control.

### Actions

The page supports the following special actions:

- **Level of Service.** Use this action to display the output of the level of service algorithms for the batch control.
- **View Source.** If the batch program is shipped with the base package, you can use the adjacent icon to display the source code of this program in the Javadocs viewer.
- Batch Control Algorithms

# Batch Control Algorithms

The following table describes all the system events supported for maintenance objects.

The **Algorithms** grid contains algorithms that control important functions for instances of this batch control. You must define the following for each algorithm:

The following table describes all the system events supported by batch controls. Note that some system events are batch program specific and as such are not available to all batch controls.


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| **Error Post-Processing** | Optional | Algorithms of this type are called if the batch process fails to complete due to an error. Multiple algorithms are allowed and are executed in sequence order. Refer to Error Post-Processing Logic for more information. |
| **File Upload** | Optional | Algorithms of this type are only applicable to plug-in driven background processes and are used to upload all records for a file.<br>This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used. |
| **Level of Service** | Optional | Algorithms of this type are called to determine the Level of Service provided by a batch control. Multiple algorithms are allowed. Refer to Assessing Level of Service for more information. |
| **Parameter Validation** | Optional | Algorithms of this type are only applicable to plug-in driven background processes and are used to validate batch specific parameters. The algorithm is invoked prior to the select and process records plug-ins.<br>This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used. |
| `Post-Processing` | Optional | Algorithms of this type are called after all threads are complete. Multiple algorithms are allowed and are executed in sequence order. Refer to Post-Processing Logic for more information. |
| **Process Record** | Optional | Algorithms of this type are only applicable to plug-in driven background processes and are used to process a specific record.<br>This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used. |


Copyright © 2026, Oracle and/or its affiliates.

| System Event | Optional / Required | Description |
|---|---|---|
| **Select Records** | Optional | Algorithms of this type are only applicable to plug-in driven background processes and are used to define the SQL to use to select the records to process.<br>This is a single algorithm plug-in spot. If multiple algorithms are plugged into the batch control, the one with the highest sequence is used. |

> **Note**
>
> **You can add new system events**. Your implementation may want to add additional batch control oriented system events. To do that, add your new values to the customizable lookup field `F1_BATCH_CTRL_SEVT_FLG`.

# Defining File Integration Types

This section describes the functionality supported for viewing and maintaining file integration types.

Refer to Extract Record Processing for an overview of the file integration type functionality provided with plug-in driven background processes.

To view or add a file integration type, open **Admin**, and then **Integration**, and then **File Integration Type.**

Once a file integration type record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **File Integration Records** zone lists the record definitions for this integration, in hierarchical order. Use the zone actions to add, maintain or view a file integration record. Press the **Record Layout** icon to view a pop-up display of the record's schema.

# Defining File Integration Records

This section describes the functionality supported for viewing and maintaining file integration record definitions.

Refer to Extract Record Processing for an overview of the file integration type functionality provided with plug-in driven background processes.

To view or add a file integration record, open **Admin**, and then **Integration**, and then **File Integration Type** and navigate to it from the **File Integration Records** zone.

The **File Integration Record** zone displays details about the selected record. Use the **View Record Types** zone action to review the complete list of record types included in the file integration type configuration.


Copyright © 2026, Oracle and/or its affiliates.

# On-Line Batch Submission

The on-line batch submission portal enables you to request a specific background process to be run. When submitting a background process on-line, you may override standard system parameters and you may be required to supply additional parameters for your specific background process. After submitting your background process, you may use this portal to review the status of the submission.

The following topics further describe logic available for on-line submission of background processes.

- Batch Submission Creates a Batch Run
- Jobs Submitted in the Background
- Email Notification
- Running Multi-Threaded Processes
- Submitting Jobs in the Future
- Lifecycle of a Batch Job Submission
- Granting Access To Batch Submission
- Submission User and Method
- Batch Job Submission Portal

## Batch Submission Creates a Batch Run

When you request a batch job to be submitted from on-line, the execution of the desired background process will result in the creation of a batch run. Just as with background processes executed through your scheduler, you may use the Batch Run portal to view the status of the run, the status of each thread, the run-instances of each thread, and any messages that might have occurred during the run.

> **Note**
>
> Your on-line submission record is assigned a status value so that you may know whether your job has been submitted and whether or not it has ended, however, it will not contain any information about the results of the background process itself. You must navigate to the Batch Run portal to view this detail.

## Jobs Submitted in the Background

When you save a record on the batch job submission page, the batch job does not get submitted automatically. Rather, it saves a record in the batch job table. A special background process will periodically check this table for pending records and will execute the batch job. This background process will update the status of the batch job submission record so that a user can determine when their job is complete.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> At installation time, your system administrator will set up this special background process to periodically check for pending records in the batch job submission table. Your administrator will define how often the system will look for pending records in this table.

It should be noted that this special background process only submits one pending batch job submission record at a time. It submits a job and waits for it to end before submitting the next pending job.

> **Note**
>
> If you request a batch job to be run multi-threaded, the special background process will submit the job as requested. It will wait for all threads to complete before marking the batch job submission record as **ended**. Refer to Running Multi-threaded Processes for more information.

## Email Notification

If you wish the system to inform you when the background process completes, you may supply your email address. The email you receive will contain details related to the batch job's output; similar to the job results you would see from your batch scheduler.

The email includes information from the product installation to help users that may be working in and submitting background processes in more than one environment.

- It includes the product name and product version for the product in the installed products list that is marked as the one to display.
- If a domain name message is configured in the installation messages, this information is also included.

> **Note**
>
> This assumes that during the installation process, your system administrator configured the system to enable email notification. Your administrator may also override the amount of detail included in the email notification.

> **Note**
>
> A batch control parameter may be used to indicate if the email should only be sent in the case of an error or success or both. Refer to Parameters Supplied to Background Processes for more information.

## Running Multi-Threaded Processes

Many of the system background processes may be run multi-threaded. When submitting a background process on-line, you may also run a multi-threaded process. The fields Thread


Copyright © 2026, Oracle and/or its affiliates.

Count and Thread Number on the batch submission portal control the multi-threaded process requests:

- To run a multi-threaded process, indicate the number of threads in **Thread Count** and enter `0` in the **Thread Number**. If a previous run ended with one or more threads in error, a new submission may be created to restart all the threads that were in error. This submission would result in new Batch Thread Instance records added for each thread that previously ended in error. Note that if the previous run was marked to "Do Not Restart", then a new submission will result in a new Batch Run.
- In the situation where a previous run resulted in one or more threads in error, you may use the Thread Number to indicate a specific thread to restart. This is only valid if there is a previous run found that is not marked "Do Not Restart". This will result in a new Batch Thread Instance created for the specific thread in the previous run.
- To run a process as a single thread, enter Thread Count = `1` and Thread Number = `1` (or `0`). This will execute the background process single-threaded.

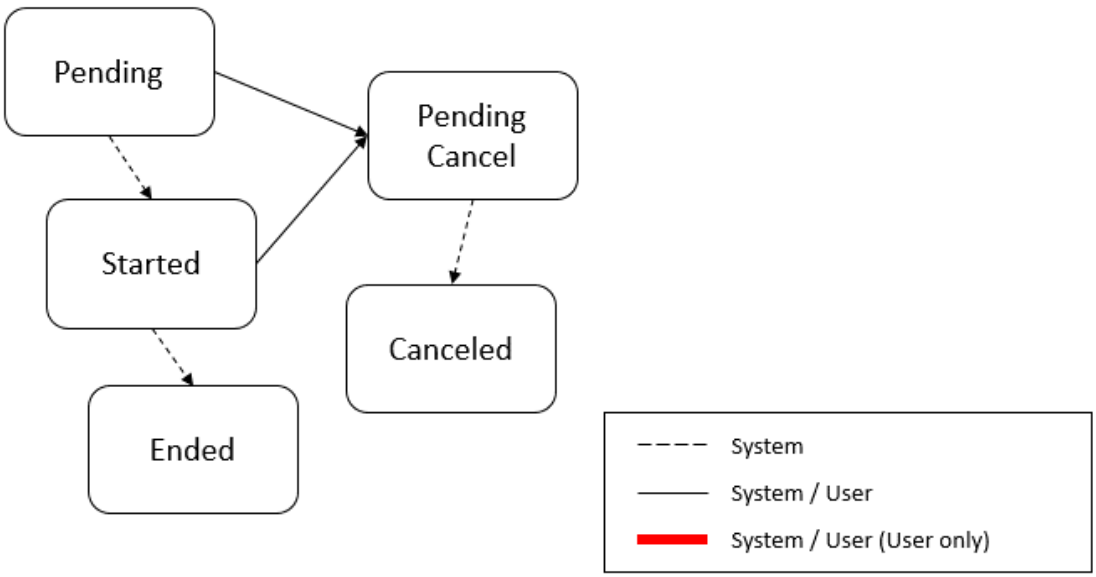
> **Note**
>
> When running a multi-threaded process, the special background process will wait until all threads are complete before marking the batch job submission record as **Ended**.

## Submitting Jobs in the Future

If you wish to request a batch job to be submitted in the future, you may do so when creating your batch job submission record by entering a future submission date. The special background process, which looks for pending records in the batch job submission table, will only submit batch jobs that do not have a future submission date.

## Lifecycle of a Batch Job Submission

The following diagram illustrates the lifecycle of a batch job submission record.





Copyright © 2026, Oracle and/or its affiliates.

**Pending** – Records are created in **Pending** status. Records in this state are put in a queue to be submitted.

**Started** – Once a pending record has been submitted for processing, its status will be changed to **Started**.

**Pending Cancel** – Users may request the cancellation of a pending record to prevent the batch job from being submitted. In addition, users may request to cancel a started batch job. The system will look for records in this status and look for batch runs that may need to be canceled for this batch job. Once it is confirmed that any batch runs are stopped, the status changes to **Canceled**.

**Ended** – When the batch job has finished processing, its status will be changed to **Ended**. Note that batch jobs in **Ended** status may have ended in error. You will need to navigate to the Batch Run portal to determine the status of the batch run.

## Granting Access To Batch Submission

Every batch control must reference an application service. When you link a batch control to an application service, you are granting all users in the group the ability to submit the associated batch job for execution.

> **NOT_SUPPORTED**
>
> Refer to The Big Picture Of Application Security for information about granting users access rights to an application service.

When the batch control is added to the batch job table, the system checks if both the user submitting the job and the user ID recorded on the batch submission record have access rights. Application security does not apply when viewing a batch control or an associated batch run.

> **Note**
>
> Base batch controls will be associated with base owned application services by default. Implementations will need to ensure the appropriate user groups are linked to these application services.

## Submission User and Method

Batch jobs track submisssion details using the following attributes.

The **Submission User** indicates who initiated the batch job and may differ from the batch user who controls permissions and auditing during the actual execution of the job. This value is not populated for timed batch jobs or batch jobs submitted via a scheduler.

The **Submission Method** indicates how the batch job was initiated. Valid methods include:

- **Online** - populated when a user manually creates or manually duplicates a batch job submission.
- **Generated** - populated by algorithms that submit a batch job and by 'initiator' batch jobs that submit other batch jobs.


Copyright © 2026, Oracle and/or its affiliates.

- **Scheduled** - populated by the DBMS scheduler. Note that for other schedulers, it is not advised to use this value as there may be logic and validation related to records created with this value that may not apply to other schedulers.
- **Timed** - populated by the batch daemon that submits timed batch jobs.
- **Other** - populated when no other value is provided.

Additional values may be added by your implementation. Refer to the lookup `SUBMIT_METH_FLG` for a list of the submission methods supported by your system.

## Batch Job Submission Portal

To view a submitted batch job, use **Menu**, and then **Tools**, and then **Batch Job Submission**. You are brought to a query portal with options for searching for a Batch Job Submisssion entry.

Once a batch job submission entry has been selected, you are brought to the maintenance portal to view the selected record.

The inline help provides information about several fields. Refer to Parameters Supplied to Background Processes for more information about the fields on this page.

Refer to On-Line Batch Submission for detailed explanations about many feaures of online submisssion.

**Actions**

This page has three special actions.

- Using the **Duplicate** button opens a window with the selected batch job submisssion details displayed. These fields, including the batch parameter values, may be overidden. Pressing **Save** will create a new Batch Job Submission entry.
- Use the **Rerun** button to rerun the selected batch job. A window opens displaying the details of the batch job submisssion to be rerun. These fields, including the batch parameter values, may be overidden. Pressing **Save** will create a rerun Batch Job Submission entry.

> **Note**
>
> Rerun is only intended to be used for background processes that download information that belongs to given run number. It is typically only required if you need to download an historical run (rather than the latest run).

- **Cancel** is only applicable for batch job submisssions that are in **Pending** or **Started** status and do not have a submission method of **Scheduled** . The action is not visible for other statuses. Refer to Lifecycle of a Batch Job Submission for more information.

# Tracking Batch Processes

The topics in this section cover the tools provided for reviewing the status and performance of batch processes. Refer to Defining Batch Controls for more information on how batch control codes are defined.

- Batch Run
- Batch Day Dashboard


Copyright © 2026, Oracle and/or its affiliates.

# Batch Run

The batch run portal shows the execution status of a selected batch process. The information includes thread status summary, details about each thread, its instances, and any error To Do Entries and messages (informational, warnings, and errors) that might have occurred during the run. It also includes information about files that may have been created by the process along with various types of analytical reports.

You can access the portal from the **Main Menu**, and then **Tools**, and then **Batch Run**. You are brought to a query portal with options for searching for a specific batch run. Once a batch run has been selected you are brought to this portal to view the selected record.

The following points should help understand the concept of threads:

- Many batch jobs cannot take advantage of your hardware's processing power when they are run singularly. Rather, you'll find that a large percentage of the CPU and/or disk drives are idle.
- In order to minimize the amount of idle time (and increase the throughput of your batch processes), we allow you to set up your batch processes so that multiple instances of a given batch job are executed at the same time. For example, in Oracle Utilities Customer Care and Billing when you schedule the **billing** process, you can indicate that multiple parallel instances should be executed (rather than just one instance). You'd do this so that the processing burden of creating bills for your customers can be spread over multiple processes.
- We refer to each parallel execution of a batch process as a "thread".
- Statistics and information messages are displayed in respect of each thread. Why? Because each thread is a separate execution and therefore can start and end at different times.

Refer to Understanding Background Processes for more information.

The following zones may appear as part of the portal's **Main** tab page.

- **Batch Run.** This zone provides information about the selected batch process, including a high level summary of key details such as threads status, processing record counts and more.
- **Batch Jobs.** This zone lists one or more batch job submission records that were used to execute the batch run. Typically a batch run is associated with a single job submission record but multiple may exist when some of the threads needed to be restarted. When a single job record exists the batch parameters specified at submission time are listed on the Batch Parameters zone. When multiple jobs exist, select a job to view its parameters.
- **Batch Parameters.** This zone lists the parameters specified as part of a selected batch job submission. This zone is hidden by default when more than one job exists for the batch run.

The following zones may appear as part of the portal's **Threads** tab page.

- **Batch Run Threads.** This zone allows you to filter and review thread instances information. When an instance is associated with messages, you may click on the message count to review the message in the Batch Instance Message zone. You may click on the Thread Summary action link to review a summary by thread.
  About Log files:
  - **Availability**. The ability to download batch log files depends on whether the application properties file has been configured with the location of the log files and the log files associated with the batch run are still available.


Copyright © 2026, Oracle and/or its affiliates.

  - **Security Access.** Only users that have security access to the **Download** access mode for the batch run application service can download log files.
  - **Compression.** The log files are compressed and include a *.gz extension. Different browsers treat this type of file differently. Some browsers may automatically decompress the file as part of the download and they are viewable in any text viewer by changing the extension to one the text viewer recognizes. However, some browsers download the compressed file and a user will need to unzip the file prior to viewing.
  - **Batch Run Logs.** The **Download stdout** and **Download stderr** zone action links may be used to download log files at the batch run level. These files may exist for processing that occurs before the thread execution. The links are visible when the corresponding file is available and the user has security rights to download files.
  - **Batch Thread Logs.** If the log files associated with a batch thread are still available and the user has security rights to download files, the **Download stdout** and **Download stderr** icons are visible for instances of that thread.

- **Batch Instance Messages.** This zone lists any messages reported by a selected thread instance. The zone is hidden by default. Refer to Processing Errors for information about these types of errors.
- **Batch Run Thread Summary.** This zone summarizes thread information across instances based on the "Accumulate All Instances" indication on the batch control record. The zone is visible upon request by using the Thread Summary link on the Threads zone.

The following zones may appear as part of the portal's **Errors** tab page.

- **Batch Instance Messages.** This zone allows you to filter and review messages across all thread instances. Refer to Processing Errors for information about these types of errors.
- **Batch Run To Do Entries.** If the background process has been enabled to create To Do entries for object specific errors, information about the To Do entries are displayed in this tab. This zone displays a summary of all To Do entries created by this batch process, grouped by status and error message. The zone is only visible when such entries exist.
- **Sample To Do Entries.** This zone displays a sample of To Do Entries for a specific status and application message selected in the Batch Run To Do Entries zone. The zone is hidden by default.

The following zones may appear as part of the portal's **Files** tab page.

- **Batch Run Files.** This zone lists file names created by the batch process.

The following zones may appear as part of the portal's **Threads Analysis** tab page.

- **Threads Analysis.** This zone includes several analytical reports around thread performance.

The following zones may appear as part of the portal's **History** tab page. This tab is only available when the maintenance object supports lifecycle definition.

- **Previous 20 Runs - Records Per Seconds.** This zone compares the throughput of the previous 20 runs.
- **Previous 20 Runs - Processed Records.** This zone compares the number of records processed by the previous 20 runs.
- **Previous 20 Runs.** This zone lists the previous 20 runs along with key details about each run.
- **Concurrent Batch Runs.** This zone lists other batch processes that ran at the same time as the current batch run. This information is sourced from the batch snapshot tables and is therefore only available when snapshot data is kept up to date.


Copyright © 2026, Oracle and/or its affiliates.

### Actions

The page supports the following special actions:

- **Do Not Attempt Restart.** If the **Run Status** is **Error**, the system will attempt to restart this run when you attempt to execute the Batch Control. In most situations, this is exactly what you want to happen. However, there are rare situations where you do not want the system to execute a given batch run (e.g., if this run is somehow corrupt and you cannot correct the data for whatever reasons). If you want the system to skip the execution of a batch run (and proceed to the next run), click on the **Do Not Attempt Restart** action.
- **Attempt Restart.** Use this action to revert back to the standard restart functionality.

## Batch Day Dashboard

This portal provides a high level summary of the status, duration and other details of the batch jobs that ran on a given business date. An implementation may introduce additional zones to this portal as needed.

Use **Menu**, and then **Dashboards**, and then **Batch Day Dashboard** to navigate to the portal.

### Batch Runs By Business Day

This zone provides a summary by business date of the number of batch runs completed, in error, and in progress, together with the earliest start time and latest end time of the jobs on that day. sThe zone is sorted in descending order by business date.

> **Note**
>
> For performance reasons, the zone is configured to restrict the number of rows retrieved by the query. The 'Results' number will indicate how many rows have been retrieved. Use the Next and Previous buttons to retrieve additional dates or return to a prior set of dates.

The following filters are provided:

- **Batch Business Date** allows you to select a particular business date.
- **Process Date** allows you to restrict the batch runs included in the summary to those where the processing date is within the start and end dates for the run.
- **Exclude Reruns** allows you to exclude batch reruns from the summary.

> **Note**
>
> The values selected in these filters will be applied as hidden filters to the other zones in the portal.

When a specific business date is broadcast from the summary, the following zones will appear as part of the portal's **Main** tab page.

### Batch Day By Status

This zone summarizes the batch runs for the business date by their status, displayed in the form of a pie chart. Selecting a specific slice of data on the chart broadcasts its value to the


Copyright © 2026, Oracle and/or its affiliates.

other zones on the portal allowing them to filter their data accordingly. Use the **Reset** action to reset the filter set by this zone.

### Batch Day By Category

This zone summarizes the batch runs for the business date by their batch category, displayed in the form of a pie chart. Selecting a specific slice of data on the chart broadcasts its value to the other zones on the portal allowing them to filter their data accordingly. Use the **Reset** action to reset the filter set by this zone.

### Batch Jobs Summary

This zone displays details of the individual batch runs for the broadcast business date. The zone is sorted in ascending order by start date/time and batch control.

> **Note**
>
> For performance reasons, the zone is configured to restrict the number of rows retrieved by the query. The 'Results' number will indicate how many rows have been retrieved. Use the Next and Previous buttons to retrieve additional rows or return to a prior set of rows.

The following filters are provided:

- **Batch Control** allows you to select a particular batch control to review. This filter supports wildcard searching.
- **From Start Date/Time** restricts the batch runs displayed to those where the start date time is on or after this date and time.
- **To Start Date/Time** restricts the batch runs displayed to those where the start date time is on or before this date and time.
- **From End Date/Time** restricts the batch runs displayed to those where the end date time is on or after this date and time.
- **To End Date/Time** restricts the batch runs displayed to those where the end date time is on or before this date and time.
- **Batch Control Type** restricts the batch runs displayed to the selected type.

The following columns are displayed initially:

- **Start Date/Time** is the start time of the run.
- **Batch Control** is the batch control ID of the batch run.
- **Description** is the description of the batch control. Click the hyperlink to navigate to the Batch Control portal.
- **Batch Run** is the run number of the batch job. Click the hyperlink to navigate to the Batch Run portal.
- **Status** is the status of the batch run.
- **End Date/Time** is the end time of the run. The current date and time may be displayed if the batch job is still in progress.
- **Duration** is the time taken for the run, formatted as Days:Hours:Minutes:Seconds.
- **Threads** is the total threads for the batch run.


Copyright © 2026, Oracle and/or its affiliates.

Click the **Column Configuration** icon in the zone header to show the Column Configuration Area in which additional columns can be added to the results. The additional columns include the following:

- **Batch Category** is the category of the batch run.
- **Batch Control Type** is the type of the batch run.
- **Do Not Attempt Restart** indicates whether the run has been marked not to be restarted.
- **Rerun Nbr** is the rerun number of this batch job.

# Requests

Requests enable an implementer to design a process using the configuration tools and standard business object monitoring techniques. They are intended to support use cases where processing should be carried out in background, but is not performed often enough that a dedicated batch process is needed.

An example of such a process might be to send an email to a group of users that summarizes the To Do entries that are assigned to them.

The following topics provide more details of requests and their uses.

- Understanding Requests
- Setting Up Request Types
- Maintaining Requests

## Understanding Requests

There are two objects that work together to define a request process:

- The request type which defines configuration related to the request.
- The request record which manages the request process and captures information required to perform the request.

### Request Type

The request type business object defines the transactional business object for requests of that type. It may also define parameters to be used by the request, such as to do types for user notification at different points in the request process. In addition, the request type lifecycle may be used to automatically create requests of this type, using a monitor algorithm and the standard monitoring batch control.

### Requests

The request object is the object that is responsible for managing a request. A separate request business object must be defined for each use case that follows a unique lifecycle. The business object schema defines the details that are captured during the process. In addition, the business object lifecycle must include at least one algorithm that is responsible for performing the request processing, as an enter plug-in on the appropriate state.

### Managing Requests

The product provides a portal for searching and maintaining request objects.

Refer to Maintaining Requests for more information.


Copyright © 2026, Oracle and/or its affiliates.

- Designing a New Request
- Configuring Requests


# Designing a New Request

The following sections highlight important considerations in setting up a new Request.

### Defining the Request Process Details

When defining a new request object, you must first determine what information needs to be captured to perform the request. Some static details, such as to do types to be used for notifications may be configured on the request type BO. Others, such as the details of the records affected may need to be recorded in the request record. These elements are defined in the BO data area of the request using standard UI hint syntax.

In addition, there may be several steps in processing a request, depending on the complexity of the use case. A request may require approval or review before being processed. These steps are modeled using the request business object lifecycle.

The system provides a 'root' business object for data correction requests (`F1-DataCorrectionRoot`) that may be used as an example of a multi-step process. Your specific product may supply additional request business objects.

### Previewing a Request

Users often want to preview a sample of the results of the request before submitting for processing. The system provides the following to support preview functionality.

- A special BO option of **Request Preview Service Script**. This script retrieves the information that is displayed when a user asks for a preview of a request.
- The `F1-DataCorrectionReqActions` map fragment. This map fragment includes support for a preview action using script `F1ReqPrvwAct`. This script invokes the request preview script configured on the request BO, using the preview script schema to generate the UI and display the results. The map also supports hiding the edit and preview actions based on flags set in the request BO.

### Performing Request Processing

The assumption is that an enter algorithm is responsible for request processing. It is common for requests to be processed in background rather than real-time. The standard technique is to use a deferred monitor to transition the request to the status that has the algorithm plugged in. The `F1-OrphanRecordDeletion` BO uses this technique.

### Request Security

Some users may not be authorized to submit certain requests. When adding a new request, the list of available request types is restricted to types where the user has security access to the related request business object's application service. This may affect the choice of application service and the additional configuration required for a new request business object.

# Configuring Requests

If your product supports request use cases that you are planning to implement, all you need to do is add a request type record using the business object for that type and define the appropriate configuration for your business requirements.

The following topics provide information about the request types supplied in the framework.



Copyright © 2026, Oracle and/or its affiliates.

- Orphan Record Deletion
- Configurable Data Correction
- To Do Summary Email


## Orphan Record Deletion

The product supports a data correction method to delete orphan records from a given table. Orphan records are child table records where the parent record no longer exists.

The following points describe the functionality at a high level:

- A request is created to capture the table from which the orphan records are to be deleted. The assumption is that prior investigation has identified the target table.
- Back end logic generates the SQL to delete the orphan records and provides counts of the rows to be deleted.
- When the request is submitted, a batch process is started that will run the SQL in background. Backups are taken prior to performing the deletion.
- When the request is completed, a separate request may be created that cleans up the backup tables from the database.

This functionality is supported by configuration and logic supplied in the base business objects Orphan Record Deletion (**F1-OrphanRecordDeletion**) and Orphan Record Deletion Type (**F1-OrphanRecDelType**). The following sections describe key steps in the request lifecycle.

### Capturing the Deletion Details

The base product provides a business service **F1-SelectOrphanRecordsSQL** that constructs the SQL to select orphan records from a target table. The business service returns additional SQL statements for the target's child and grandchild tables, if applicable. The product also provides the business service **F1-CountRecords** which counts the orphan records based on the SQL generated by **F1-SelectOrphanRecordsSQL**.

The Orphan Record Deletion BO invokes these business services in a post-processing algorithm and records the details on the request record.

### Approving and Submitting the Request

Once a deletion request is captured, there are several key steps to perform.

- Deletion request records are created in **Pending** state so that the SQL statements and record counts can be reviewed before proceeding.
- If the deletion details are correct, the request can be transitioned to **Ready for Approval**. A to do entry is created to notify potential approvers. The request must be approved by someone other than the user who created the record. The notification to do entry type and role are configured on the Orphan Record Deletion Type.
- The request must be approved before it can be submitted for processing. If the Orphan Record Deletion Type indicates that record counts must be verified before deletion, a warning will be issued at approval time if there is a record mismatch.
- Once the request is in the **Approved** state, it can be submitted. The **Submitted** state is configured to be processed by a batch monitor so that record deletion is carried out in background. The monitor batch job is submitted automatically. When record deletion is complete, the request transitions to the **Processed** state so that the results can be reviewed.



Copyright © 2026, Oracle and/or its affiliates.

- If the deletion could not be completed successfully, the request will enter the **Error** state and a to do entry is created. The error to do entry type and role are configured on the Orphan Record Deletion Type.

#### Completion and Clean Up

The record deletion process includes steps to take backups of the target tables in case the results are not as expected. A separate request may be created to handle deletion of unwanted backup tables at an appropriate time. If the Orphan Record Deletion Type specifies a clean up request type, a request record of that type is created when the deletion request enters the **Complete** state. The clean up functionality is supported by configuration and logic supplied in the base business objects Backup Table Cleanup (**F1-BackupTableCleanup**) and Backup Table Cleanup Type (**F1-BackupTableCleanupType**).

## Configurable Data Correction

Implementers can use plug-in driven batches to perform some types of one-off data corrections but the process lacks the built in approval step supported by the request based data correction flows. Moreover, each custom plug-in driven batch involves creating a few custom algorithms, scripts and a batch control, all generated for a one-off update.

The product supports a simplified way to request, approve and perform plug-in driven based data corrections.

The following points describe the functionality at a high level:

- A data correction request is created to capture the following details:
  - An **SQL** query that selects the records to correct.
  - Either a **business object** or a **service script** to perform the data correction for each selected record.
  - The number of threads to submit the data correction batch process with.
- When the request is approved and a user submits it for processing, a batch job is created for the specific request. The batch job uses the query on the request to select the records to correct and calls the corresponding business object or script to perform the update for each record. The request remains in the 'Submitted' status while the batch process runs.
- When the batch completes, it updates the corresponding request to the 'Processed' status, which then creates a monitoring To Do Entry for a user to review the results of the batch. The user marks the request as complete and if further corrections are needed the user can create another request with different correction details as needed.

The following sections provide more information about this tool.

#### Request Type Setup

A "Data Correction via Batch" (**F1-DataCorrectTypeByBatch**) Request Type business object is provided to define a request type record for creating such requests. To enable this functionality, a request type should be created using this business object.

The request type should reference the "Configurable Data Correction" (**F1-DataCorrectPlugIn**) as the request business object. This business object is a subclass of the data correction root business object and as such leverages the same approval based lifecycle support.

#### Data Correction Implementation

The following is a suggested description for a one-off data correction procedure:


Copyright © 2026, Oracle and/or its affiliates.

- If the change is not a simple update of fields using a business object, implement a service script that gets the prime key of a selected record and uses standard scripting logic to perform the update. The script can be tested using the Test action on the Script page.
- Use any tool to compose and test an SQL that selects the records from a primary table of a maintenance object to correct.
- If the change is a simple update of non-list fields on a business object, note the xpath of the primary key fields and the xpath of each element that needs to be updated on that business object. Adjust the SQL to set the xpath element name as the alias of each field.

  This is an example of a data correction SQL that selects some batch control records and assigns each with a new "next batch run number". The "batchControl" and "nextBatchNumber" aliases are xpath references to the corresponding fields in the F1-BatchControlPhysicalBO business object.

  ```
  select batch_cd "batchControl",
  case
  when batch_cd = '...' then <new next batch run number>
  ...
  else ...
  end "nextBatchNumber"
  from ci_batch_ctrl
  where batch_cd in ('.... list of batch control records to update ... ')
  ```

  > **Note**
  >
  > The SQL is provided for illustration purposes only.

- If the change is not as simple, implement a service script that gets the fields from a selected record and uses standard scripting logic to perform the update.
  - The script's schema must include the **F1-DataCorrectionProcessRecord** data area. It is via this data area that your script gets the selected field from each row to update. The structure of this data area is similar to the plug-in driven **Process Record** plug-in spot.

    All selected fields are provided to your script in the 'selectFields' list of the data area. In addition, the primary key of the selected record may also be provided via the 'mo', and 'pk1-5' elements, if the 'MAINT_OBJ_CD', and 'PK_VALUE1-5' designated aliases are used in your query.

    For example:

    ```
    select 'F1-FACT' MAINT_OBJ_CD, fact_id PK_VALUE1, ...
    from f1_fact ...
    ```

  - The script can be tested using the 'Test' action on the Script page.
- Create a 'Configurable Data Correction' Request using the corresponding Request Type. Specify on the request the SQL and the business object or script to use for data correction.
- Progress the request through approval and submit for processing when ready.
- Review the data once the submitted batch is complete.

### Approving and Submitting the Request

Once a data correction request is captured, there are several key steps to perform.


Copyright © 2026, Oracle and/or its affiliates.

- If the deletion details are correct, the request can be transitioned to **Ready for Approval**. A to do entry is created to notify potential approvers. The request must be approved by someone other than the user who created the record. The notification to do entry type and role are configured on the Request Type.
- The request must be approved before it can be submitted for processing.
- Once the request is in the **Approved** state, it can be submitted. The **Submitted** state is configured to be processed by a plug-in driven batch process that uses details from the request to perform the deletion in background. The batch job is submitted automatically. When batch job completes, the request transitions to the **Processed** state so that the results can be reviewed.
- A user marks the request as complete and if further corrections are needed the user can create another request with different correction details as needed.

## To Do Summary Email

The base product includes a sample request process that sends an email to users that have incomplete To Dos older than a specified number of days.

### Configuring the Email Request Type

The following configuration tasks are required to use this process:

- Define an Outbound Message Type. The business object usually defined for the Outbound Message Type is **F1-EmailMessage.**
- Define an External System that contains the Outbound Message Type in one of its steps. In the configuration determine if the communication is through SOA, batch, or real-time processing method when sending the email notification. Refer to Outbound Messages for more information about configuration needed for the different processing methods.
- Create a Request Type using the `F1-TodoSumEmailTyp` business object.
- Configure the Request Type with the outbound message type and external system.
- Provide values for the threshold, email subject and email address.
- If you want emails of this type to be sent periodically, enter a value for the frequency parameter.

### Previewing the Email Request

When adding or editing a To Do email request, the following actions are possible:

- Click **Save** to submit the request.
- Click **Cancel** to cancel the request.
- Click **Preview** to see a sample of records that satisfy the selection criteria for this request. This information is displayed in a separate map. In addition, the map displays the total number of records that will be processed when the request is submitted. From this map you can click **Save** to submit the request, **Back** to adjust the parameters, or **Cancel** to cancel the request.

### Submitting the Email Request

When an email request is saved, the job is not immediately submitted for real time processing. The record is saved with the status **Pending** and a monitor process for this record's business object is responsible for transitioning the record to **Complete**.


Copyright © 2026, Oracle and/or its affiliates.

As long as the record is still **Pending**, it may be edited to adjust the parameters. The preview logic described above may be repeated when editing a record.

The actual work of the request, such as generating an email, is performed when transitioning to **Complete** (using an enter processing algorithm for the business object).

## Setting Up Request Types

This portal is used to view and maintain request types.

You can access this portal from **Admin**, and then **General**, and then **Request Type**. You are brought to a query portal with options for searching for a specific request type. Once a request type has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to the embedded help text for more information on the Request Type attributes.

Refer to Understanding Requests for background information about defining request types.

## Maintaining Requests

This portal is used to view and maintain requests.

You can access this portal from **Main Menu**, and then **Tools**, and then **Request**. You are brought to a query portal with options for searching for a specific request. Once a request has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding Requests for more information.

# Attachments

Some implementations may require that attachments be available from the application. These attachments can be stored in the Attachment table and then linked to other objects if applicable.

- Attachment Overview
- Configuring Your System for Attachments
- Maintaining Attachments

## Attachment Overview

The following topics provide additional information regarding attachment functionality.

**Attachment Types**

The system supports several different attachment content types, for example:

- PDF Document
- Excel Spreadsheet
- Jpeg Image
- Text Document


Copyright © 2026, Oracle and/or its affiliates.

The attachment data itself may be text or binary. When storing the data in the application however, it is stored as text information only. As a result, the upload of an attachment that is a binary type requires a conversion prior to storing the data. When viewing the attachment, the data is converted again for display.

The system also supports attachments that reference a URL to a document hosted in an external system. When viewing the attachment, the URL is launched to access the related document in the external system.

Each type of attachment is defined using an attachment business object. The business object includes configuration defining the supported file extensions, whether the data is binary or not and the content type that represents the type of data for the attachment.

> **Note**
>
> To view the attachment business objects provided with the base product, navigate using **Admin**, and then **System**, and then **Business Object**, and then **Search** and search for business objects related to the Maintenance Object for Attachments (`F1-ATCHMT`). You may set any of these business objects as not allowing new instances to prevent their use for creating attachments.

### Owned Attachments

Attachments can be either ‘owned’ or ‘common’. An owned attachment is one that is related to a specific record. For example, the specific test results for a given device can be uploaded and linked to that device or to its test records. These types of attachments are typically uploaded and maintained via the object that owns it.

### Common Attachments

Common attachments are ones that are uploaded independent of any transaction in the system. They can be used for general system or company information. Or they can be linked to more than one transaction. For example, instructions for performing a certain type of task can be uploaded as an attachment and linked to a task type where those instructions are relevant. These types of attachments are uploaded and maintained in the central Attachment portal. Objects that may refer to the attachments may link the attachments via characteristics or some other appropriate mechanism.

### Content Location

An attachment's content may reside in either of the following locations:

- **Database.** The content is stored in the database as part of the attachment record. This is also the default location when uploading a file that is not too large.
- **Object Storage.** When an uploaded file is too large to be stored in the database its content is stored in Object Storage. Refer to the "Uploading a Large Attachment File" section for more information.
- **External URL.** An attachment may reference a URL to a document hosted by an external system.

> **Note**
>
> To reference an existing Object Storae file use the file storage syntax.


Copyright © 2026, Oracle and/or its affiliates.

When sending an attachment to an external system via an outbound message or including it in an email, the attachment's content can only be embedded in the message when the content resides in the database. For all other content locations, an external facing URL to view the attachment is provided instead.

### Uploading a Large Attachment File

By default, an attachment is uploaded and stored in the database along with its entire content. The process of uploading very large content like videos etc. is often limited by memory and other resource constraints. You may configure a threshold above which content is stored in object storage and a link to the file is kept on the attachment record.

To enable this feature refer to the **attachment location** and related options in the **General System Configuration** feature configuration for more information. In a cloud installation, this configuration is required to upload large attachments due to stricter resource limits.

The following are known limitations with storing content in object storage:

- The responsibility for data recovery of attachment content shifts to the customer as part of their object storage files management.
- Custom logic if any that relies on attachment content to exist in the database will not work for attachments stored in object storage.

### Emailing Attachments

The system supports a business service that may be used by system processing to send an email. The business service `F1-EmailService` supports receiving the IDs of one or more attachments as input parameters. Refer to Sending Email for more information.

By default, an attachment's content is embedded in the email message when the content resides in the database. For all other content locations, an external facing URL to view the attachment is provided instead. When the content of the attachment is too large to be embedded in the email, a link is provided instead. Refer to the **Maximum Email Attachment Size** in the **General System Configuration** feature configuration for more information.

### External Reference

Attachment records may be added from an external system in which case, the external system could have a reference ID to capture. The attachment allows for including an external reference ID when adding an attachment internally.

### Malware Scan

The system provides a plug-in spot where on premises implementations can configure a step to scan attachment for malware. If an algorithm is provided, it is called every time attachment content is added. This is not supported in cloud implementations.

# Configuring Your System for Attachments

In order to link attachments to objects in the system, there may be some configuration or implementation required to support the link. It is possible that one or more objects in your product already support attachments out of the box. Consult the product documentation for the specific object for confirmation. For objects in the system that do not support attachments out of the box, the following sections provide some guidelines for enabling support for attachments. Contact product support for more information.


Copyright © 2026, Oracle and/or its affiliates.

### Supporting Common Attachments

The attachments themselves are created / uploaded using the attachment portal. Refer to Maintaining Attachments for more information.

If your implementation has a use case where one or more common attachments may be linked to an object (and the object does not already support this functionality), the object may need to be extended to capture the attachments.

- If the object includes a characteristic collection, this is a recommended way to capture attachments. A characteristic type should be defined for each type of attachment. The characteristic type should be a foreign key type and should reference the Attachment FK reference. The characteristic entity collection should include the object that the common attachment will be linked to.
- Most characteristic collections are sequence based characteristics and would support multiple entries for the same characteristic types, if multiple attachments are applicable.
- If the object to support the attachments is governed by a business object, the implementation must extend the business object to define one or more appropriate elements used to capture the attachments. If only one attachment of a certain type is allowed, a single flattened characteristic may be used. If multiple attachments of a certain type are allowed, the BO schema may define a “flattened list” exposing the sequence and the characteristic type.
- If the object is maintained on a “fixed page” with a generic characteristic collection, no additional configuration is needed to allow users to link attachments to that object.

### Supporting Owned Attachments

When creating an attachment for a specific record, the attachment itself captures the information about the related record, namely its maintenance object code and its primary key. For these types of attachments, no configuration is needed on the related business object to capture the attachments, as was the case with common attachments.

However, it is recommended to configure the user interface of the related object so that the owned attachments can be viewed and maintained from that page. To do this, you may use the generic attachment zone provided by the product: `F1-ATTCHOWN`.

Note that your product may already have support for viewing and maintaining owned attachments on one or more if its base delivered portals.

### Configuring Size Limit

The system allows for configuration of a size limit for uploading attachments. This is optional but the recommendation is to set this to a reasonable value so as to limit the ability to upload unreasonably big files. To configure this, navigate to Feature Configuration. Search for an existing feature configuration with the Feature Type **General System Configuration**. If one does not exist, create a feature configuration of this type. Select the **Maximum Attachment Size** option and define an appropriate value.

You may also configure the system to upload very large attachments to Object Storage. To enable this functionality, set the **Attachment Location** option in the same feature configuration to define the file location in object storage. Configure the **Attachment Threshold Size** option to set the size limit above which attachment data is stored in object storage as files and not in the database.

If your organization performs many concurrent attachment uploads, you may optionally set the **Maximum Attachment Aggregate Size** option to override the system limit as needed.


Copyright © 2026, Oracle and/or its affiliates.

When the content of the attachment is too large to be embedded in an email, an external facing URL to view the attachment is provided instead. Refer to the **Maximum Email Attachment Size** in the **General System Configuration** feature configuration for more information.

### Defining a New Attachment Type

As mentioned, the product provides support for several content types. If your implementation needs to support attachments for a content type not currently supported, create a new business object copying the configuration of an existing attachment business object.

Configure the following option types for the BO:

- **Binary** indicates whether the attachment data must be converted from binary format. Binary attachments are stored in the database as text, and are then converted back to the original format when retrieved.
- **Content Type** represents the browser's mime type of the attachment.
- **Supported File Extension** specifies the valid file extensions for the content type.

Once the business object is defined, it is ready for use.

### Supporting File Name Exceptions

By default, the system prevents attachment file names from having special characters. If there is some reason that an implementation needs to define an attachment with special characters, there is a system property setting that may be configured to relax this validation. Contact your system administrator for support.

### Enabling Malware Scan

> **Note**
>
> This functionality is only available for on premises implementations. It is not supported for cloud implementations.

The product provides a malware scan algorithm in the base product that uses ClamAV. ClamAV itself is not provided as part of the product. To use the base algorithm, your implementation needs to get ClamAV and define system properties for the host and port where ClamAV may be called. The algorithm, called `F1-ATCHM-WS` should be plugged into the installation options - algorithms collection using the system event **Attachment Malware Scan**

If your implementation wants to use a different algorithm for malware scanning, you may deliver your own algorithm and plugit into the installation algorithm collection.

# Maintaining Attachments

This section describes the functionality supported for viewing and maintaining attachments.

Navigate using **Admin**, then **General**, the **Attachment**. You are brought to a query portal with options for searching for common attachments.

Once an attachment has been selected, you are brought to the maintenance portal to view and maintain the selected record.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The base search options for the attachments query only support searching for common attachments. Owned attachments may also be viewed on the attachment maintenance portal, but a user may only drill into the attachment maintenance from the maintenance portal of the "owning" entity.

The Attachment zone provides basic information about an attachment, including the ability to upload the file and to view an uploaded file. Viewing an attachment will typically launch a download of the attachment file, following standard browser behavior.

- Adding Attachments

## Adding Attachments

Common attachments may be added from the attachments portal (or via the standard menu path). In addition, your product may support attachments associated with specific records ("entity owned attachments") which may also provide the capability to add attachments.

In both cases, when adding an attachment, you are prompted for either a file to upload or an external URL to a document that resides in another system. When uploading a file then once a file is chose, the system determines the appropriate business object to associate with the attachment based on the file extension. Typically one and only one business object is found at which point you are prompted to provide the Attachment Name. (Your specific product may also require additional information at this time). Fill in the details and save.

Please note the following:

- If no business object is found for the uploaded file's file type, an error is issued. This type of file is not currently supported as an attachment.
- If multiple business objects are found, the user must choose the appropriate one. This should be rare.
- The ability to add URL based attachments is enabled by default but may have been disabled by your organization.
- The attachment's file extension is validated against the base delivered allow list F1-FileExtensionAllowList, defined as a managed content record.
- The system ensures that the attachment's file name does not contain special characters and uses only standard UTF-8 supported characters.

# Monitoring and Analytics

This chapter describes various tools provided in the product to support system monitoring including tools related to reporting, monitoring and analytics.

- Understanding Batch Analytics
- Reporting Tool Integration
- Health Assessment Options
- Service Health Check
- Measuring Performance
- Capturing Statistics


Copyright © 2026, Oracle and/or its affiliates.

- Capturing Product Use Metrics

# Understanding Batch Analytics

This section explains how to use batch analytics for strategic analysis of jobs, threads, and workload performance.

- Batch Job Analytics
- Batch Thread Analytics
- Batch Workflow Analytics

## Batch Job Analytics

The Batch Job Analytics portal illustrates a set of metrics aligned with execution of batch processes at the job level. This portal is useful for summarization of the state and work performed by a batch control over a selected date range.

> **Note**
>
> To use these analytics, you must populate the batch snapshot must be populated, on a regular basis, using the batch job **F1BRANSN - Batch Run Analytics Snapshot Update.**.

> **Note**
>
> Not all zones in this portal support broadcast as it is not appropriate in all situations. Unless indicated assume broadcast is not available from the zone.

> **Note**
>
> At the top of each tab in this portal, the selected batch control and date range is shown.

### Batch Job Analytics Search

To use this portal you must first select the batch control and the period you want to analyze using the Batch Job Analytics search from the search dropdown. To help find the necessary batch control, you can use a variety of filters to refine your search.

> **Note**
>
> **Date Range is required** At a minimum a date range must be specified. This must be either the predefined period or a custom date range. For custom date ranges, it is recommended to limit the range to analyze less than a thousand (1000) values for performance and readability reasons.


Copyright © 2026, Oracle and/or its affiliates.

Once the appropriate filters are populated a list of batch controls is returned with the number of executions in the snapshot for the specified date range and other filters specified. To start the analysis, select the desired batch control from the list.

### Batch Job Analytics - Main

The Main section of the analytics summarizes the state, submission source, performance, and volumes processed by the desired batch control over a date range. This includes analysis in the following areas:

- The Batch Job Status Summary zone summarizes the state of all executions of the desired batch control over the desired date range within the snapshot. In an ideal situation this graph should show a high percentage of complete (successful) executions.
- The Batch Job Submission Summary zone summarizes the method used to submit the desired batch control over the desired date range within the snapshot. In an ideal situation, batch jobs are submitted the using a small amount of methods to maximize efficiency.
- The Batch Job Elapsed Time Summary zone shows the minimum (best) elapsed time, average elapsed time and maximum (worst) elapsed time for the desired batch control over the desired date range within the snapshot. All elapsed times are shown in seconds, with the hover text translates the time into DD:HHMI:SS format. In an ideal situation, all the values should be relatively the same. Large deviations in values may be due to volume or inefficiencies.
- The Batch Job Volume Summary zone shows the minimum (lowest) volume, average volume and maximum (highest) volume for the desired batch control over the desired date range within the snapshot. Volume is expressed as the total number of objects processed by the job. In an ideal situation, all the values should be relatively the same. Large deviations in values may be due to business fluctuations or inefficiencies.

### Batch Job Analytics - History

The History section of the analytics shows detail information for job level elapsed time and volumes for the desired batch control over the desired date range within the snapshot.

Each analysis includes additional reference information such as the average value and the confidence interval to use. Values above the confidence interval should be considered as outliers and require further analysis.

This includes analysis in the following areas:

- The Batch Job Elapsed Time History zone shows the job level elapsed time, in seconds (the hover text translates the time into DD:HHMI:SS format) for each batch run for the desired batch control over the desired date range within the snapshot. This can highlight fluctuations in elapsed time of data processed by the batch job over time.
- The Batch Job Volume History zone shows the total number of objects processed by each batch run for the desired batch control over the desired date range within the snapshot. This can highlight fluctuations in volumes of data processed by the batch job over time.

### Batch Job Analytics - Submission

The Submission section of the analytics shows analysis of submission information periods applicable to the desired batch control over the desired date range within the snapshot.

This includes analysis in the following areas:

- The Batch Job Submission History zone shows the number of submissions per day over the for the desired batch control over the desired date range within the snapshot. This can


Copyright © 2026, Oracle and/or its affiliates.

highlight fluctuations in how often the desired batch control is submitted. Ideally the number of submissions should be consistent.

- The Batch Job Submission By Day Of Week zone is an analysis of the total submissions by each day of the week to help detect any fluctuations in submission patterns for the desired batch control over the desired date range within the snapshot. Ideally the number of submissions should be consistent.
- The Batch Job Submission By Hour Of Day zone is an analysis of the total submissions by each hour for the desired batch control over the desired date range within the snapshot. This zone is provided to help detect any fluctuations in submission patterns. Ideally the number of submissions should be consistent.
- The Top Users Submitted Job zone summarizes the top 20 users that have submitted the desired batch control over the desired date range as expressed in the snapshot. The total number of submissions by those users is shown for the desired batch control over the desired date range within the snapshot. This zone is primarily provided to help detect unusual patterns by individual users as necessary.

#### Batch Job Analytics - Detail

The Detail section of the analytics lists all batch runs for the desired batch control over the desired date range as expressed in the snapshot. This represents the data that was used as part of the analysis used in the previous sections. It is possible to navigate to the Batch Run Portal for individual runs for further information. This list can be further filtered for status and submission method used.

## Batch Thread Analytics

The Batch Thread Analytics portal illustrates a set of metrics aligned with execution of batch processes at the thread. This portal is useful for summarization of the state and work performed by the threads for a batch control over a selected date range. This portal is useful for jobs that are run multi-threaded.

> **Note**
>
> To use these analytics, you must populate the batch snapshot must be populated, on a regular basis, using the batch job **F1BTANSN - Batch Thread Analytics Snapshot Update.**.

> **Note**
>
> Not all zones in this portal support broadcast as it is not appropriate in all situations. Unless indicated assume broadcast is not available from the zone.

> **Note**
>
> At the top of each tab in this portal, the selected batch control and date range is shown.


Copyright © 2026, Oracle and/or its affiliates.

### Batch Thread Analytics Search

To use this portal you must first select the batch control and the period you want to analyze using the Batch Thread Analytics search from the search dropdown. To help find the necessary batch control, you can use a variety of filters to refine your search.

> **Note**
>
> **Date Range is required** At a minimum a date range must be specified. This must be either the predefined period or a custom date range. For custom date ranges, it is recommended to limit the range to analyze less than a thousand (1000) values for performance and readability reasons.

Once the appropriate filters are populated a list of batch controls is returned with the number of executions and the maximum thread limit used in the snapshot for the specified date range and other filters specified. To start the analysis, select the desired batch control from the list.

### Batch Thread Analytics - Main

The Main section of the analytics summarizes the state, submission source, performance, and volumes processed by the desired batch control over a date range. This includes analysis in the following areas:

- The Batch Thread Status zone summarizes the state of all executions of the desired batch control over the desired date range within the snapshot. In an ideal situation this graph should show a high percentage of complete (successful) executions.
- The Batch Thread Reliability Summary zone calculates the total reliability rating for the desired batch control over the desired date range within the snapshot.
- The Batch Thread Limit History zone shows the thread limit specified for each batch run for the desired batch control over the desired date range within the snapshot. Ideally the thread limit used should be consistent. Large deviations in values may be due to volume or inefficiencies.

### Batch Thread Analytics - Reliability

The Reliability section of the analytics focuses on the reliability of individual job runs for the desired batch control over the desired date range within the snapshot.

The analysis of threads introduces a reliability rating related to the success rate of threads for an execution. The percentage of successful threads over total thread count for each execution is used for the rating using the following criteria:

- If all threads (100%) are successful for a batch run then the rating is **High**.
- If the thread failure rate is less or equal to 1% of the total number of threads then the rating is **Good**.
- If the thread failure rate is less or equal to 5% of the total number of threads then the rating is **Fair**.
- If the thread failure rate is above 5% of the total number of threads then the rating is **Poor**.

The reliability analysis includes analysis in the following areas:

- The Batch Thread Reliability History zone shows the reliability rating for each run for the desired batch control over the desired date range within the snapshot. The color and size of the entries in the graph illustrate the reliability and percentage of successful threads.


Copyright © 2026, Oracle and/or its affiliates.

- The Batch Thread Reliability Details zone lists the individual batch runs with their associated reliability rating for the desired batch control over the desired date range within the snapshot. You can navigate to the Batch Run Portal for additional information.

**Batch Thread Analytics - Differential**

The Differential section of the analytics shows analysis of submission information periods applicable to the desired batch control over the desired date range within the snapshot.

This includes analysis in the following areas:

- The Batch Thread Time Differentials zone shows the difference between start and end time for the desired batch control over the desired date range within the snapshot as expressed as a wost case differential. Each differential is shown in seconds. If threads start at the same time and end at the same time then the differential is zero (0). Positive differentials mean differences in start and end times of threads.
- The Batch Thread Volume Differentials zone shows the difference between the smallest volume and largest volume for the threads within each run against the average volume. Each differential is shown as number of records that are different compared to the average volume. A reference line is shown to show the relative difference in low and end volumes. In an ideal situation each thread should work on the same amount of records. Variations means variance in processing volume and may result in different thread elapsed times.

**Batch Thread Analytics - Retries**

> **Note**
>
> If this zone is blank, then there were no automatic retries for the desired batch control over the desired date range within the snapshot.

The Retries section of the analytics lists the total retries per batch run. If the number of retries is above zero, in a consistent way, then there may be some underlying issues that need investigation.

> **Note**
>
> Retries are only performed for specific conditions. Refer to the cloud service documentation for additional information.

**Batch Thread Analytics - Zero Processing**

> **Note**
>
> If this zone is blank, then there were no zero processing threads found for the desired batch control over the desired date range within the snapshot.

While most threads have records to process, it is possible that a thread is allocated no work to do. This is inefficiency as resources are allocated to start an end the thread at a minimum. This zone highlights the number of threads per batch run with zero processing for the desired batch control over the desired date range within the snapshot. If the number is consistently high then consider altering the thread limits or thread numbers to re-balance the workload.


Copyright © 2026, Oracle and/or its affiliates.

### Batch Thread Analytics - Interquartile Range Analysis

Interquartile analysis is a statistical method used to understand the spread and distribution of a set of data by focusing on the middle 50% of values. By dividing the data into four equal parts, it helps identify the range where most data points fall, making it easier to spot any unusual or extreme values (outliers). This approach is helpful for quickly assessing data consistency and detecting trends, enabling you to make more informed decisions based on typical, rather than exceptional, results.

The Batch Thread Inter Quartile Analysis zone shows the inter-quartile analysis for threads across batch runs for the desired batch control over the desired date range within the snapshot.

To understand the analysis there are different metrics calculated to provide reference:

- The **median quartile** represents the median between the worst and best thread elapsed time. This metric is calculated to provide a reference to assess other metrics against. In an ideal situation all the threads would have this elapsed time.
- The **upper quartile whisker** and **lower upper quartile whisker** are calculated at the 75% and 25% confidence interval, respectively, from the median. These whiskers represent the acceptable extremes for elapsed times of threads. There proximity to the median illustrates whether the threads are finishing
- The **high within quartile** and **low within quartile** calculates the thread elapsed time within the whiskers. The proximity of these metrics to the whiskers indicates if threads are ending early and/or late consistently.
- The **worst thread elapsed time** and **best thread elapsed time** are used as reference for all the above metrics. If these values are above their respective whiskers then they are consider outliers
- The difference between the whispers is known as the **Inter Quartile Range (IQR)**. The higher the IQR value the more the variation in the thread elapsed times. If these is little variation, then these values are close to the median.

It is divided into three types of analysis, as different graphs on the zone:

- The first analysis is focusing on the upper quartile against the worst performing threads. The relationships between these metrics will illustrate the extent of any variation in the threading and the potential presence of threads that overrun.
- The second analysis is focusing on the lower quartile against the best performing threads. The relationships between these metrics will illustrate the extent of any variation in the threading and the potential presence of threads that finish too early.
- The last analysis is focusing on the IQR value. If you see a small IQR, it means thread execution times are consistent. A large IQR suggests there is significant variation.

## Batch Workflow Analytics

The Batch Workflow Analytics portal illustrates a set of metrics aligned with execution of batch processes at the workload level. This portal is useful for ;looking at workload at the threadpool level to determine load profiles over a particular day for a particular threadpool.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> To use these analytics, you must populate the batch snapshot must be populated, on a regular basis, using the batch job **F1BTANSN - Batch Thread Analytics Snapshot Update.**.

> **Note**
>
> Workload Analytics use a "workload minute" to measure capacity. Each thread running in any minute uses one workload minute. If multiple threads run in the same minute, each one adds to the total. This makes it easy to see how much capacity is used during a day, since the maximum available is the number of allowed threads multiplied by 1,440 minutes (the total minutes in a day) for each thread pool.

### Batch Workload Analytics Search

To use this portal you must first select the batch control and the period you want to analyze using the Batch Workload Analytics search from the search dropdown. To help find the necessary threadpool and date, you can use a variety of filters to refine your search.

> **Note**
>
> **Date Range is required** At a minimum a date range must be specified. This must be either the predefined period or a custom date range. For custom date ranges, it is recommended to limit the range to analyze less than a thousand (1000) values for performance and readability reasons.

Once the appropriate filters are populated a list of threadpools and dates is returned with the total workload minutes for the combination of threadpool and date in the snapshot for the specified date range and other filters specified. To start the analysis, select the desired threadpool and date combination from the list.

### Batch Workload Analytics - Main

The Main section of the analytics summarizes workload minutes for the desired threadpool and desired date for every two minute interval. This zone shows how busy the desired threadpool was at any point in the desire date. You filter the information with additional filters as necessary.

Clicking on any point in time will broadcast the time interval and the Batch Workload Detail will show the batch jobs and threads that were started, running and/or complete in that interval. You can navigate to the Batch Run Portal for additional information.

### Batch Workload Analytics - Summary

The Summary section of the workflow analytics provides summary level analytics related to the total workload minutes in the desired threadpool and desired date. This includes analysis in the following areas:

- The Batch Workload Summary by Category zone summarizes total workload minutes by batch category. This can be used to appreciate the distribution of workload for the desired threadpool and desired date.


Copyright © 2026, Oracle and/or its affiliates.

- The Batch Workload Summary by Submission zone summarizes total workload minutes by submission type. This can be used to appreciate the distribution of how the workload is submitted for the desired threadpool and desired date.

**Batch Workload Analytics - Hour Analysis**

If the minute by minute analysis is too detailed, the Batch Workload Hour Analysis zone shows workload by hour of the day for the desired threadpool and desired date.

# Reporting Tool Integration

This section describes how to configure your third party reporting tool and how to define your reports in the system to enable users to submit reports online.

- The Big Picture Of Reports
- Configuring The System To Enable Reports
- Sample Reports Supplied with the Product
- How To Define A New Report

## The Big Picture Of Reports

The topics in this section describe the approach for designing and defining your system reports. Note that the product includes an out-of-the-box integration with BI Publisher. However it is possible to use the reporting objects to integrate with a different third party tool.

- Integration with BI Publisher
- How To Request Reports
- Viewing Reports

### Integration with BI Publisher

Your DBMS, your product, and BI Publisher can work together to produce reports. You may choose to use a different reporting tool, but this may not be a trivial effort. This section provides high-level information about some of the business requirements that are being solved with the reporting solution.

**Multi-Language and Localization Support**

The integration supports a multi-language implementation and supports different localization settings.:

- All labels, headings and messages are defined using field and message meta-data in the application, which support multiple languages.
- The appropriate font, size, and layout are based on the requested report and the user's language.
- Dates and numbers are formatted as per the user's display profile.
- Currency based numbers are formatted as per the currency definition from the product
- Requesting Reports from The System
- Overview of the Data - BI Publisher


Copyright © 2026, Oracle and/or its affiliates.

## Requesting Reports from The System

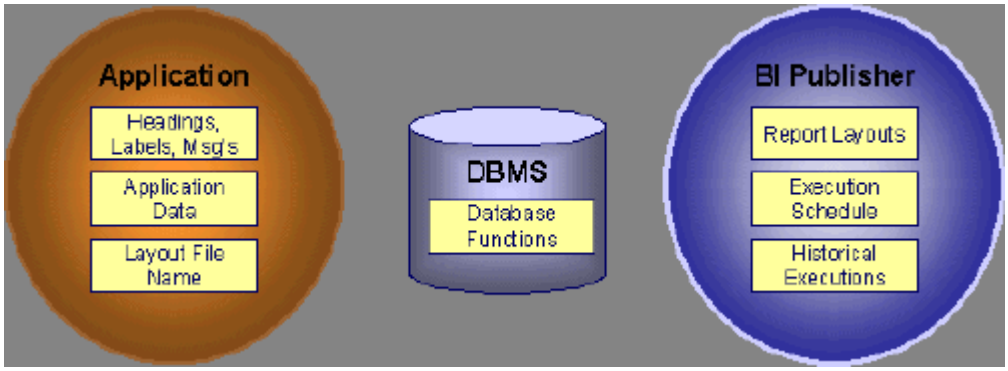
Although reports are rendered in your reporting tool, users must be able to request ad-hoc reports from within the system (assuming users have the appropriate security access).

- The prompts for the input parameters must be shown in the user's language
- Users should be able to use the standard search facilities to find parameter values
- Plug-ins can be optionally used to cross-validate input parameters
- Application security must authorize ad-hoc report requests

## Overview of the Data - BI Publisher

The following diagram provides an overview of where data is stored for your reports for integration with BI Publisher.

**Figure Application and BI Publisher**




The application contains:

- The application data that appears on your reports.
- The language-specific headings, labels and messages on your reports.
- The layout file name to be used for the report.

BI Publisher contains:

- How your reports look.
- Information about scheduled reports and reports that have already run.

The DBMS contains the SQL used to retrieve the data on your reports (residing in database functions).

> **Note**
>
> BI Publisher can be configured to retrieve data via a service call. Because every business object can be read via a service call, elements that reside in an XML based column can be displayed on reports produced using BI Publisher. See your product's *Installation Guide* or *Optional Products Installation Guide* for information on this configuration.


Copyright © 2026, Oracle and/or its affiliates.

## How To Request Reports

A user may request an ad hoc report from within your product:

- A report submission page enables a user to choose the desired report and enter the parameter values for the report
- The user must be granted security access to the report
- The request is passed to the reporting tool real time. Refer to Configure The System to Invoke BI Publisher for more information.
- The reporting tool generates the report and displays it in a new browser window

The reporting tools' scheduler creates reports (as per your schedule). This function is entirely within the reporting tool. No scheduling functions reside within your product.

A user can request an ad-hoc report from within the reporting tool. Note, the user's ID must be supplied as a parameter to the report in order for the user's profile to be used to format dates and numbers

## Viewing Reports

As described above, ad-hoc reports requested from within your product are displayed immediately after they are generated in a new browser window

If your reporting tools supports it, the Report History page may be configured to open your tool's report execution history page and request a view of this report.

> **Note**
>
> The Report History page currently does not display historical reports for BI Publisher.

# Configuring The System To Enable Reports

- Configuring BI Publisher Reports
- Defining Reporting Options
- Defining Report Definitions

## Configuring BI Publisher Reports

This section contains topics specific about configuring the product to interoperate with BI Publisher.

- Configure the System to Invoke BI Publisher Real-time
- Batch Scheduling in BI Publisher

### Configure the System to Invoke BI Publisher Real-time

The base product provides an installation algorithm plug-in spot called Reporting Tool. This plug-in spot should contain an algorithm that invokes the third party reporting tool real-time.

For BI Publisher, the system provides an algorithm type called `F1-BIPR-INV`, which invokes BI Publisher.


Copyright © 2026, Oracle and/or its affiliates.

These algorithms rely on information defined in the Reporting Options table: the reporting server, reporting folder and the user name and password for accessing the reporting tool. The values in the reporting options should have been set up when the system was installed. Contact your system administrator if there are any problems with the values defined on the reporting options.

To use the algorithm types to invoke BI Publisher, perform the following steps:

- Create an algorithm for the appropriate algorithm type.
- On the installation options, add an entry to the algorithm collection with an algorithm entity of **Reporting Tool** and indicate the algorithm created in the previous step.

### Batch Scheduling in BI Publisher

For many of your reports, you probably want the report to be produced on a regular basis according to a scheduler. The reporting solution relies on the BI Publisher software to provide the batch scheduler functionality. Refer to BI Publisher documentation for details about configuring the batch scheduler.

## Defining Reporting Options

The reporting options are provided as a mechanism for defining information needed by your reporting solution. The base product uses the reporting options to define information needed to access the reporting tool from within the system using the algorithm defined on the installation option.

Navigate to this page using **Admin**, and then **Reporting**, and then **Reporting Options**.

**Description of page**

**Reporting Folder** defines the shared folder where reports are stored.

**Reporting Server From Browser** defines the URL of the web application where the reporting tool is installed. For example, using BI Publisher, the format is: `http://<BI Publisher Server>:<port>`.

> **Note**
>
> **Customize Options**. The reporting options are customizable using the Lookup table. This field name is `RPT_OPT_FLG`. The reporting options provided with the system are needed to invoke the reporting tool. If your implementation requires other information to be defined as reporting options, use the lookup table to define additional values for the reporting option flag.

**Where Used**

This information is used by the reporting tool algorithm on the installation option to invoke the reporting tool software.

Implementations may use reporting options to record other information needed for their reporting tool.

## Defining Report Definitions

For each report supplied by your installation, use the report definition page to define various attributes of the report.


Copyright © 2026, Oracle and/or its affiliates.

- Report Definition - Main
- Report Definition - Labels
- Report Definition - Parameters

## Report Definition - Main

Navigate to this page using **Admin**, and then **Reporting**, and then **Report Definition**.

> ⚠ **Caution**
>
> Important! If you introduce new report definitions, you must prefix the report code with **CM**. If you do not do this, there is a slight possibility that a future release of the application could introduce a new system report with the name you allocated.

**Description of page**

Enter an easily recognizable **Report Code** and **Description** for each report. Use the **External Reference ID** to define the identifier for this report in your external reporting tool.

Define an application service to enable users to request submission of this report online or to view report history for this report. Once you define an application service for each report, use Secured Objects and Application Services to define which users may access this report.

> ⓘ **Note**
>
> **Access Mode**. The access mode for application services related to reports must be set to **Submit/View Report**.

If you have more than one parameter defined for your report and you wish to perform cross-validation for more than one parameter, provide an appropriate **Validation Algorithm**.

Enter a **Long Description** to more fully describe the functionality of this report. This information is displayed to the user when attempting to submit the report online or when viewing history for this report.

For BI Publisher, if you want to use one of the sample reports provided by the system, but with a different layout, indicate the layout to use for the report in the **Customer Specific Font/ Layout** field and BI Publisher uses this information instead. The name for base report layout is <report code>_Base. For example, a base layout for CM_TODO is named CM_TODO_Base.

## Report Definition - Labels

Navigate to this page using **Admin**, and then **Reporting**, and then **Report Definition** and go to the **Labels** tab.

> ⓘ **Note**
>
> **Company name and logo.** Note the company name used as a title in the sample reports is defined as a message on the installation options. For information about installing the company logo, refer to your product's *Installation Guide* or the *Optional Products Installation Guide*.


Copyright © 2026, Oracle and/or its affiliates.

**Description of Page**

In order to provide multi-language capability for each report, the labels used for the report must support multiple language definitions. For each label used by your report, indicate a unique **Sequence** and the Field used to define the **Label**. The label defined here should be the same label that is defined in your report layout defined in the external reporting tool.

When rendering an image of the report, the external reporting tool retrieves the appropriate label based on the language used for the report.

### Report Definition - Parameters

Navigate to this page using **Admin**, and then **Reporting**, and then **Report Definition** and go to the **Parameters** tab **.**

**Description of Page**

The **Parameters** scroll contains one entry for every parameter defined for the report. The following fields display:

**Parameter Code** is the identifier of the parameter. This must correspond to the parameter definition in the reporting tool.

**Required** indicates that a value for the parameter must be defined when submitting the report.

**Sort Sequence** must match the parameter order defined in the reporting tool's report. It is also used when displaying the list of parameters on the report submission page.

**Characteristic Type** indicates the characteristic type used to define this parameter.

**Default Value** is option and if populated is displayed to the user when the report is chosen on the report submission page.

**Description** is a brief description of the parameter. This description is used when displaying the parameter on the report submission page.

**Long Description** is a detailed description of the parameter. This description is used on the report submission page when the user requests more information for a given parameter.

## Sample Reports Supplied with the Product

Depending on your specific product, there may be sample reports provided that your organization may use as they are or as a starting point for creating a new report. The following sections provide an overview of the sample reports along with instructions on how to use one of the sample reports in your implementation environment.

- How to Use a Sample Report Provided with the System

### How to Use a Sample Report Provided with the System

If you would like to use any of the sample reports, you need to perform some steps to be able to execute them in an implementation environment. This section walks you through the steps needed.

- Steps Performed at Installation Time


Copyright © 2026, Oracle and/or its affiliates.

### Steps Performed at Installation Time

Refer to the *Installation Guide* or *Optional Products Installation Guide* for instructions for setting up and configuring your product and reporting tool to use the sample reports provided with the system. The following steps are described there.

- Setting up the stored procedures used by the sample reports.
- Defining the company title and logo used by the sample reports. Note the company name used as a title in the sample reports is defined as a message on the installation options.
- Defining a user for integration with your product.
- Publishing the sample reports in BI Publisher.

Contact your system administrator to verify that the above steps have occurred.

## How To Define A New Report

- Use a Sample Report as a Starting Point
- Publishing Reports in BI Publisher
- Designing Your Report Definition

## Use a Sample Report as a Starting Point

- Make a copy of the report and save it in an appropriate directory. Prefix the new report name with **CM**.
- Review the stored procedure(s) used for this report. Refer to the installation guide for information about where the stored procedures should be defined. If you want to change the data that is being accessed, copy the stored procedure, prefixing the new stored procedure with **CM**. Make the appropriate changes in the new version of the stored procedure. Contact your database administrator to find out the procedure for creating a new stored procedure.

> **Note**
>
> **Performance considerations**. When designing a stored procedure, you must consider the performance of the report when executed. Consult your database administrator when designing your database access to ensure that all issues are considered.

> **Note**
>
> **Defining Messages**. The stored procedures provided with the system use messages defined in message category 30. If your new stored procedures require new messages, use message category 90000 or greater, which are reserved for implementations.

- Review the parameters used by the report. Make appropriate changes to the parameters required by the report. This affects how you define your report. Refer to Designing Parameters for more information.


Copyright © 2026, Oracle and/or its affiliates.

- Determine whether or not you require cross validation for your report parameters. If any cross validation is necessary, you should design an appropriate validation algorithm to be executed when requesting a report in your product. Refer to Designing Validation Algorithms for more information.

> **Note**
>
> **Cross Validation for On-line Submission Only**. The cross validation algorithm is only executed for ad-hoc report submissions via your product. If you submit this report through your reporting tool, this algorithm is not executed.

- Review the labels used by the report. Labels and other verbiage are implemented in the sample reports using a reference to the field table in the system. This enables the report to be rendered in the appropriate language for the user. For any new report label you require, you must define a new field entry. Refer to Designing Labels for more information.
- Review the layout of the report and make any desired changes based on your business needs.

When you have finished designing and coding your new report in your reporting tool, you must do the following in order for it to be usable:

- Publish the report in BI Publisher. Refer to the documentation for this products for details about publishing a report. Refer to Publishing Reports in BI Publisher for configuration information specific to publishing a report for integration with your product.
- Define the report. Refer to Designing Your Report Definition for more information.

## Publishing Reports in BI Publisher

Please refer to the documentation for BI Publisher for more information about publishing a report in this system. The remaining topics in this section provide information about settings needed to ensure that the report is accessible using BI Publisher.

- BI Publisher Database Access
- Verify BI Publisher User Access Rights

### BI Publisher Database Access

When publishing a report in BI Publisher, you are asked for database logon information. The logon user name and password must be the user name and password that has access to the database functions related to this report in your database.

### Verify BI Publisher User Access Rights

To verify the user's access rights to folders in BI Publisher:

- Open the BI Publisher Enterprise Security Center.
- Check that the role for the user has access to the appropriate report folders.

For more information, refer to the "Understanding Users and Roles" section in the Oracle Business Intelligence Publisher User's Guide.


Copyright © 2026, Oracle and/or its affiliates.

# Designing Your Report Definition

When adding a new report, you must define it in the system to allow users to request ad-hoc reports from on-line and to take advantage of the multi-language provisions in the system. The following topics illustrate the steps to take to correctly configure your report definition.

- Designing Main Report Definition Values
- Designing Characteristic Types
- Designing Parameters
- Designing Validation Algorithms
- Designing Application Services
- Designing Labels

## Designing Main Report Definition Values

Refer to field description section of the report definition main page for information about defining general information about the report.

For the validation algorithm, preliminary steps are required. Refer to Designing Validation Algorithms for more information.

For the application service, preliminary steps are required. Refer to Designing Application Services for more information.

## Designing Characteristic Types

The parameter tab on the report definition page uses Characteristic Type to define the report parameters. For each report parameter that you plan to use, you must define a characteristic type.

You do not need a unique characteristic type for each report parameter. For example, if Start Date and End Date are parameters your report, only one **Report Date** characteristic type needs to be defined. This characteristic type would be used on both date parameters.

Each characteristic type to be used as a report parameter must indicate a characteristic entity of **Report**.

To illustrate the characteristic type definitions, let's look at the sample report Tax Payables Analysis. It needs the following parameters: From Date, To Date, GL Account Type Characteristic Type and Account Type value.

> **Note**
>
> **Account Type Parameters**. The tax payables report must find general ledger entries that have posted to a certain distribution code. In order to find the appropriate distribution code, the report expects each distribution code to be defined with a characteristic indicating its GL account type (for example, **Revenue**, **Asset**, etc.) The report needs to know the characteristic type used to define this entry.

To support the required parameters, the following characteristic types are needed.

| Char Type | Description | Type | Valid Values | Char Entities |
|---|---|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | | | | |
|---|---|---|---|---|
| CI_DATE | Date Parameter | Ad-hoc | (Uses validation algorithm to validate proper date entry) | Report |
| CI_CHTYP | Characteristic Type | FK Reference | CHAR_TYP | Report |
| CI_GLTY | GL Account Type | Pre-defined | A- Asset, E-Expense, LM-Liability/ miscellaneous, LT-Liability/taxes, R-Revenue | Distribution Code, Report |

Highlights for some of the above settings:

- We have defined a characteristic type for defining a characteristic type. This is to allow the user to indicate which Char Type on the Distribution Code is used for the GL account type. This is an FK reference type of characteristic.
- The GL account type characteristic type is referenced on both the Distribution Code entity and the report entity.

## Designing Parameters

Your report definition parameters collection must define a unique parameter entry for each parameter sent to the reporting tool. The sequence of your parameters must match the sequence defined in your reporting tool.

Continuing with the Tax Payables Analysis report as an example, let's look at the parameter definitions.

| Parameter Code | Description | Char Type | Default Value |
|---|---|---|---|
| P_FROM_DT | From Date | CI_DATE | N/A |
| P_TO_DT | To Date | CI_DATE | N/A |
| P_CHAR_TYPE | Account Type Characteristic | CI_CHTYP | CI_GLTY |
| P_TAX_ACCTY_CHAR | Account Type Char Value for Tax Related GL Account | CI_GLTY | LT-Liability/taxes |

Highlights for some of the above settings:

- The from date and to date parameters use the same characteristic type.
- The characteristic type parameter is defined with a default value pointing to the GL account type characteristic type.
- The GL account type parameter defines the liability/taxes account type as its default value.

> **Note**
>
> **User Id**. The sample reports provided by the system pass the user id as the first parameter passed to the reporting tool. It does not need to be defined in the parameter collection for the report.


Copyright © 2026, Oracle and/or its affiliates.

## Designing Validation Algorithms

When designing your report definition, determine if cross validation should occur for your collection of parameters. In the Tax Payables Analysis report, there are two date parameters. Each date parameter uses the characteristic type validation algorithm to ensure that a valid date is entered. However, perhaps additional validation is needed to ensure that the start date is prior to the end date. To do this, a validation algorithm must be designed and defined on the report definition.

The system provides a sample algorithm `RPTV-DT` that validates that two separate date parameters do not overlap. This algorithm should be used by the Tax Payables Analysis report.

If you identify additional validation algorithm, create a new algorithm type. Create an algorithm for that algorithm type with the appropriate parameter values. Plug in the new validation algorithm to the appropriate report definition.

## Designing Application Services

Application services are required in order to allow a user to submit a report on-line or to view history for a report. Define an application service for each report and define the user groups that should have submit/view access to this report.

Update report definition to reference this application service.

## Designing Labels

The system supports the rendering of a report in the language of the user. In order to support a report in multiple languages, the verbiage used in the report must be defined in a table that supports multiple languages. Some examples of verbiage in a report include the title, the labels and column headings and text such as "End of Report".

The system uses the field table to define its labels.

> **Note**
>
> **Report Definition.** This section assumes that your new report in the reporting tool has followed the standard followed in the sample reports and uses references to field names for all verbiage rather than hard-coding text in a single language.

For each label or other type of verbiage used by your report, define a field to store the text used for the verbiage.

- Navigate to the field page using **Admin**, and then **System**, and then **Field**.
- Enter a unique **Field Name**. This must correspond to the field name used in your report definition in the reporting tool and it must be prefixed with **CM**.
- Define the **Owner** as **Customer Modification**.
- Define the **Data Type** as **Character**.
- **Precision** is a required field, but is not applicable for your report fields. Enter any value here.
- Use the **Description** to define the text that should appear on the report.
- Check the **Work Field** switch. This indicates to the system that the field does not represent a field in the database.


Copyright © 2026, Oracle and/or its affiliates.

Update the report definition to define the fields applicable for this report in the **Labels** tab.

If your installation supports multiple languages, you must define the description applicable for each supported language.

# Health Assessment Options

The product provides functionality to periodically collect health assessment scores of various aspects of the application. Such mechanism has the following benefits:â€‹

- Minimize risks and costs by reporting and addressing issues promptly.
- Proactively identify common implementation issues in customer environments to address potential problems early.â€‹
- Leverage best practices that are based on proven strategies from other Oracle products to ensure consistency and efficiency.â€‹

Following points highlight some of the features of this functionality:

- The health assessment type object defines configuration related to a given assessment rule, including various criteria that determine whether it is applicable to the current environment or not.
- The health assessment object is used to capture a calculated assessment value at a given point in time along with a corresponding pass, fail or warning score.
- The Health Assessment Dashboard portal provides a high level compliance summary.
- The Health Assessment portal is provided to display information about a calculated assessment.
- A monitor batch control is used to create the health assessment records. Every time it's run, it calculates and captures another measurement for each health assessment type that is eligible for the current environment.
- In a cloud environment, the monitor batch control used to create the health assessment records is considered an internal batch job and will be run by the Oracle Utilities Cloud Service.

> **Note**
>
> Although the health assessment functionality is required for cloud implementations, implementations that are on premises may also choose to run the health assessment type's monitor batch control to capture information for viewing in the Health Assessment Dashboard portal if desired.

- Understanding Health Assessments
- Defining Health Assessment Types
- Health Assessment
- Health Assessment Dashboard

## Understanding Health Assessments

A health assessment record captures the environment's score for a given condition at a given point in time. Health assessment records are calculated and captured periodically to assess an environment health over time.


Copyright © 2026, Oracle and/or its affiliates.

Health assessment type describes a health condition that should be measured and evaluated over time along with remediation instructions for completeness. The condition may be classified as being critical or not and it is assigned a category for reporting purposes.

The definition specifies the calculation rule for measuring the condition along with settings that determine whether it is eligible for the current environment or not.

> **Note**
>
> Base delivered assessment types cannot be inactivated. They represent conditions that should be checked for product health. Your organization can mark an assessment type as hidden from report for it not to appear on the Health Assessment Dashboard portal by default. The portal allows you to toggle your view to include or exclude those as needed.

This section describes topics related to designing and reviewing health assessments.

### Eligibility Rules

A health assessment condition may only be applicable in certain environments and not others. For example, a condition that should only be checked in a production environment or is only applicable in a cloud installation, etc. By default, a health assessment type is considered eligible in any environment unless explicitly defined otherwise.

For a cloud implementation, the system looks at the applicable cloud offerings setting on the health assessment type. If that is populated with specific cloud offerings, then the assessment type is only applicable if your product is in the list. (If no applicable cloud offerings are populated, then it's considered applicable for all cloud offerings.)

The health assessment calculation algorithm, referenced on the health assessment type definition, may further determine that the assessment type is not eligible for the current environment.

### Capturing Health Assessments

The health assessment type monitor batch process, which is used to calculate and store assessments, is scheduled as an internal batch process in the cloud.

Implementations that are on premises may also choose to run the health assessment type's monitor batch control to capture information for viewing in the Health Assessment Dashboard portal if desired.

The process of calculating a health assessment of a given type is as follows:

- Basic eligibility rules are evaluated, such as whether this is a cloud or production environment, etc. If the assessment type is not eligible it is skipped.
- The calculation algorithm is called. If the algorithm indicates the assessment type is not eligible it is skipped.
- If an exception occurred while calling the algorithm, i.e. a measured value could not be calculated, the assessment score is set to **Exception** and the assessment record records the exception details.
- If the algorithm did not score the measured value, and a scoring range is defined on the assessment type, the value is scored accordingly. Scoring the measured value is required and if it cannot be scored an exception is issued.


Copyright © 2026, Oracle and/or its affiliates.

- If the algorithm did not provide the percentage to failure value, it is calculated based on the assessment score and the scoring range defined on the assessment type.
- The assessment record is stored along with any optional characteristics provided by the algorithm.

### Health Assessment Score

The measured assessment value is scored as either **Pass**, **Warning** or **Fail** to describe the severity of the condition. In addition, a **Percentage to Failure** value is also provided to indicate how close the measured value is from a failure score as a proactive alert mechanism. A pass score is set to 0% and a failed score to 100%.

The algorithm associated with the assessment type is responsible for computing a measured value. It may also provide a score for the measurement and calculate its percentage to failure value.

If the algorithm did not score the value, the system uses the scoring range defined on the assessment type to score the calculated value. A value less than or equal to the pass limit is set with a 'pass' value and if greater or equal to the 'fail' limit then is set to 'fail'. Any value is between is scored with 'warning' value. If a scoring range is not defined the assessment is scored with a 'warning' value.

For example, if an assessment type is defined with a maximum pass value of 7 and a minimum failure value of 11 then a measured value of 5 would be scored as 'pass', a value of 9 as 'waring' and a value of 11 as 'fail'. In the same way, if a percentage to failure is not set by the algorithm, the system calculates it based on the same range. A value scored as warning is set with a higher percentage to failure the close it is to the fail limit. If a scoring range is not defined the default is 0 as percentage to fail.

### Assessment Calculation Algorithm

The health assessment type definition references a processing algorithm for calculating the condition's measurement value and scoring it. If the algorithm does not score the value, settings on the assessment type may be used to do so.

Basic eligibility rules are defined on the assessment type definition. For more specific rules, the algorithm may also determine that the assessment is not eligible for the current environment and therefore should be skipped.

The algorithm may provide basic supporting details as characteristics to display along with the assessment.

The product provides generic assessment algorithms that can be used to simplify the configuration of basic assessment types. For example, an algorithm that gets a measurement value labeled as 'count' from an SQL provided as an parameter. The measured value is scored by comparing it against the scoring range defined on the assessment type. Refer to product health assessment types for examples.

### Launching A More Information Portal

An assessment type may be associated with an optional portal for reviewing specific details about an assessment. When specified, the health assessment portal provides navigation to the specific portal along with the health assessment identifier as context. This is an optional configuration and most likely needed to clarify complex assessment conditions.


Copyright © 2026, Oracle and/or its affiliates.

### Review Assessments

The Health Assessment Dashboard portal provides a high level summary of health assessment conditions. It is recommended to review this portal periodically and address highlighted issues as needed.

Health assessment types provided by the base product cannot be disabled. Customers can mark certain product assessment types to be hidden from this dashboard display. These are still assessed and collected but a user may choose to view them or not on this dashboard. The user may drill into a Health Assessment portal for more information about a given rule over time. â€‹When applicable, the user may also further navigate to the specific portal associated with the rule.

### No Custom Business Objects

Health assessment functionality relies on base product business objects for core functionality. There is no support for custom business objects nor customization of the base product's business objects. Customers can add assessment types with custom calculation rules but the assessment type and assessment entities must be based on product business objects.

## Defining Health Assessment Types

Health assessment types are mainly defined by the product but allow additional ones to be added by your organization. To view the health assessment types, open **Admin**, and then **System**, and then **Health Assessment Type.**

Use the **Health Assessment Type Query** portal to search for a health assessment type. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to Understanding Health Assessments for more information.

For information about the data defined on the product metric type, refer to embedded help.

## Health Assessment

The Health Assessment portal displays the measurement score and additional information related to a specific health assessment rule.

To view this portal, open **Main**, and then **System**, and then **Health Assessment.**

Refer to Understanding Health Assessments for more information.

The portal includes the following zones:

- **Health Assessment.** This zone displays details about a health assessment record along with guidance on how to address such issue.
- **Assessment History.** This zone lists a history of assessment values prior to the current assessment.

For more information about what each zone on the portal is reporting, refer to the embedded help.


Copyright © 2026, Oracle and/or its affiliates.

## Health Assessment Dashboard

The Health Assessment Dashboard portal provides a high level summary of health assessment scores for key aspects of your application. The information visible on this portal will depend on the product or products installed by your organization.

To view this portal, open **Main**, and then **Dashboards**, and then **Health Assessment Dashboard.**

Refer to Understanding Health Assessments for more information.

The portal includes the following zones:

- **Critical Assessments.** This zone summarizes the latest assessment for all critical assessment types by assessment score.
- **Overall Assessments.** This zone summarizes the latest assessments of all types by assessment score.
- **Assessments.** This zone lists latest assessments by assessment score.

For more information about what each zone on the portal is reporting, refer to the embedded help.

# Service Health Check

The system provides a plug-in spot on Installation, called Health Check. The algorithms plugged in here are used to check various system conditions and return details to help assess the health of the system.

### What Information is Checked?

Each algorithm is responsible for checking one type of health check condition. The condition should be defined as a valid health component type (as defined in the `HEALTH_COMP_FLG` lookup). The algorithm may check conditions across many records (referred to as Health Components) and return a status and details for each health component. For example, if the algorithm is checking the level of service for batch programs in the system, the batch control is the health component in this case. The algorithm should gather the level of service results for each batch program and return the collection of information to the calling program.

The algorithms for the plug-in spot should return the following information for each health component.

- The **Health Component Detail** should be populated with information specific to the object that was checked. For example, if the algorithm is checking batch control level of service, this will have the Batch Control code. In addition, for enabling display for details of each health component type, the algorithm should populate the maintenance object code and primary key field and value details along with the navigation option.
- The **Health Component Status Flag** and **Health Component Status Description** should be populated with a valid status code and description appropriate for the condition being checked. This information could differ based on the health component type as each health component type may have its own unique status values.
- The **Health Component Status Reason** is meant to provide supporting information about the health component's status value. For example, if the component provides an error type of status, this could have the status reason.


Copyright © 2026, Oracle and/or its affiliates.

- The **Health Component Response** is used to map the status value of the health component to a standard value defined in the lookup `HEALTH_RESPONSE_FLG`. The values are a subset of HTTP response codes. The supported values are **All Checks Successful** (200); **Non-Critical Function Degraded** (203), **No Content** (204) and **One or More Critical Functions Degraded** (500).
- The **Response Details** list is provided for health components that check several conditions as part of the health check. Each condition could return its own status. In this situation, the algorithm should use the response details list to record the individual responses and populate the overall Health Component Status Flag, Description and Reason with summary information. The suggestion is to do the following:
  - Set the status flag and response based on the details of the individual responses. Refer to the batch level of service information below for an example of this logic.
  - When the response list is populated, the health check user interface will show the overall Status Reason text along with an icon to expand the details. The suggestion is to populate this field with text like "See Results for Details".

The system provides an algorithm that checks the **Batch Level of Service** health component type. This health component type finds all the batch controls that are configured with at least one level of service algorithm and invokes the algorithms for each batch control. The business service populates the output for this health service for each batch control as follows:

- The **Health Component Detail** is populated with the Batch Control code and description. In addition, the navigation information for being able to drill into the batch control are provided and used to build the column as hypertext.
- The **Status** is populated based on whether the batch control has one algorithm or multiple. If there is one algorithm, the Level of Service lookup value returned by the algorithm is returned. If there are multiple, the system determines an overall status based on the detailed status values from each algorithm. If any algorithm returns Error, that value is returned. Otherwise, if any return a Warning, that value is returned. Otherwise Normal is returned.
- The **Status Reason** is populated based on whether the batch control has one algorithm or multiple. If there is one algorithm, the expanded text of the status reason returned by the algorithm is returned. If there are multiple, the text from message category / message number 11002 / 22001 is returned (**See Results for Details**)
- The **Response** is populated based on the value of the overall Level of Service status. It is set to **All Checks Successful** (200) when the Level of Service is **Normal** or **Disabled**; **Non-Critical Function Degraded** (203) when the Level of Service is **Warning** and **One or More Critical Functions Degraded** (500) when the Level of Service is **Error**.

**How are the Algorithms Called?**

These algorithms are called by a business service provided by the product `F1-HealthCheck`.

That service calculates an overall Health Response value based on all the details returned by all the algorithms. The values described above for the `HEALTH_RESPONSE_FLG` are the ones used.

Also note that the system provides Inbound Web Services for this business service for both SOAP and REST service calls allowing external systems to use a web service to retrieve this information.

- Health Check Configuration
- Viewing Health Check Results
- Introducing Health Check Conditions


Copyright © 2026, Oracle and/or its affiliates.

## Health Check Configuration

To enable your system to display health check information, navigate to the Installation Options - Algorithms collection and add an entry for the **Health Check** system event for each health check algorithm provided by the product that your product would like to include in your system health check assessment.

## Viewing Health Check Results

The product provides a portal that allows a user to view the detailed results of the health check business service.

Navigate using **Admin**, and then **System**, and then **Health Check** to view this portal.

The zone on the portal displays the following:

**Overall Response** is the HTTP response code returned by the business service.

The grid displays the detail of each individual component checked as part of the system health check. Refer to Service Health Check for more information about the details returned.

## Introducing Health Check Conditions

If your implementation would like to check a condition for system health that is not already provided by your product, here are the steps required:

- Define a new value in the Health Component Type lookup (`HEALTH_COMP_FLG`).
- Provide a new algorithm type, plug-in script and algorithm for the Health Check system event that checks the condition that you wish to check. Refer to the Service Health Check topic for the information that each algorithm should be supplying for the condition. Note that if you follow the pattern of the system delivered Batch Control - Level of Service algorithm, your algorithm may be designed to check conditions across many records of the same type and gather all the results.
- Plug this new algorithm into the Installation Options - Algorithms collection using the **Health Check** system event.

# Measuring Performance

Many implementations need the ability to track and view the performance of key system processes against a defined target level. The framework provides objects to allow an edge product or an implementation to calculate and display performance measures against a desired target for one or more use cases. The topics in this section provide information about what is supplied in the framework and guidelines for implementing a specific use case for batch processes. Your product may supply out of the box support for additional use cases. Refer to your product documentation for more information.

- Understanding Performance Targets
- Setting Up Performance Target Configuration
- Maintaining Performance Targets


Copyright © 2026, Oracle and/or its affiliates.

# Understanding Performance Targets

The following are examples of use cases that would be well suited for tracking as performance targets:

- Track and view the duration of key batch processes, either individually or as a group, and how they relate to a defined target.
- Monitor used and free space on a database against critical levels.
- Check the performance of individual user interface zones against a defined performance expectation
- Compare the number of web service requests made to an application against a threshold where performance may be of concern.

Framework provides functionality to define and categorize performance targets and link them to objects such as business services, zones and portals. This supports the calculation and display of the metrics against desired results.

In addition, Framework supplies out of the box support for batch process performance targets. Individual edge applications may supply more specific functionality for other use cases, if applicable.

Ideally, users should have the ability to view these performance targets on a dashboard that groups related measures. Framework provides the necessary components to achieve this for batch process performance targets.

The following sections highlight functionality supported for performance targets in the framework. Refer to the edge application product documentation for more details of other supported use cases.

- Performance Target Objects Overview
- Calculating and Displaying Performance Targets
- Performance Target Metrics and Metric Types
- Performance Target Categories and Types
- Performance Targets Define Specific Metrics
- Objects Linked to a Performance Target
- Creating Performance Target Zones

# Performance Target Objects Overview

The setup of a performance target involves a unique combination of configuration data and processing logic that calculates and displays a specific measure.

The framework performance target functionality is supported by a combination of inter-related objects, as shown below. Some of these objects will be generic for use in all performance targets while some are specific to a functional area such as batch processes.

- **Maintenance Objects** for capturing performance target types and performance target instances.
- **Extendable Lookups** to define performance target categories and performance target metrics.
- **Business Services** to calculate known metrics for a group of performance targets, such as for batch processes.


Copyright © 2026, Oracle and/or its affiliates.

- **UI Maps** to interpret the performance calculation results and display them in charts that show the comparison to the target.
- A **Zone** that serves as a template for system duplication to create specific zones related to each performance target. The zone invokes a business service to perform calculations and display performance measures in the related UI map.
- **Business Objects** to capture configuration data for a specific performance target instance and its related objects.
- Functionality to create **Zone instances** for specific performance targets based on the associated template zone.

Each edge application will deliver the following to compliment the objects delivered by the framework:

- **Portals** to group related performance target zones.
- Specific entries for the performance target category **Extendable Lookup** .

The following sections describe the combined use of these objects for performance targets in more detail.

## Calculating and Displaying Performance Targets

Performance targets are intended to be displayed in a portal using an explicit object zone. The zone parameters define both the business service used to calculate the performance metrics and the UI map that displays the results. While an individual performance target needs to reference a zone with a unique configuration that calculates a particular metric, those zones will be based on a template zone which defines the core parameters.

Framework provides a base Batch Performance Target Metric Template zone (`F1-PERFBA`) for batch process performance targets. Refer to this zone and its parameters for more information about the batch performance zone configuration and the related business service and UI map.

## Performance Target Metrics and Metric Types

The framework supports two types of metrics for performance monitoring:

- **Value based metrics** are used to record results against a specific numeric target.
- **Time based metrics** are used to track the results against a specific date and time target.

The list of valid metrics for a given performance target category and its associated performance target types is maintained using an extendable lookup. Framework uses the base business object Batch Performance Target Metric (`F1-BatchPerfTargetMetric`) to define batch process metric values. Refer to this lookup for the supported batch process metrics.

> **Note**
>
> While users are not prevented from adding new values to the lookup, the list is not intended to be extendable as new values will not be recognized by the business service that performs the base batch performance calculation logic.

Your edge product or implementation may supply other extendable lookup business objects for additional performance measurement use cases, if applicable.


Copyright © 2026, Oracle and/or its affiliates.

## Performance Target Categories and Types

There are key configuration details required by all performance targets. These are defined on two related objects.

**Performance Target Categories**

Target categories define the template zone and security setting for a group of performance measures. The list of valid categories is maintained using an extendable lookup.

The framework product supplies the business object Performance Target Category (`F1-PerformanceTargetCategory`) for this functionality. Refer to the business object description and configuration for more information.

**Performance Target Types**

Target types define the related performance target business object and the display portal for a group of performance measures. In addition, a target type references a target category which defines the associated zone details.

The framework product supplies the business object Performance Target Type (`F1-PerformanceTargetType`) for this functionality. Refer to the business object description and configuration for more information.

The framework does not deliver any standard type or category values for batch processing performance targets. Refer to your specific edge application products to verify if any standard values are delivered for the batch processes within your applications. Edge applications may also supply standard categories and types for additional performance target use cases.

## Performance Targets Define Specific Metrics

Although the types of measures and the business services and UI maps that govern how they are calculated and displayed are defined using separate objects, a **performance target** record defines the additional configuration needed to measure a specific metric and compare the result to a desired value.

There are key configuration details required by all performance target instances. These include a reference to the metric being measured, the desired target value or time, the desired result for the target and the unique zone by which this performance target will be monitored.

> **Note**
>
> The performance target maintenance object has a direct foreign key link to the extendable lookup business object and value that define the performance metric. This is an unusual pattern as extendable lookup values are normally recorded only in an object's schema. The pattern has been adopted to allow the description of the metric to be displayed in the performance target maintenance portal.

A given **performance target type** may require additional details for its calculations. For example, the framework batch process performance target defines additional details to restrict the measurement to specific batch processes that have executed within a given time frame. These details are configured on the performance target business object. Refer to the embedded help text on the business objects supplied by the framework and your edge applications for more information.


Copyright © 2026, Oracle and/or its affiliates.

The **performance target type** also defines the business object to use when creating the resulting performance target record.

## Objects Linked to a Performance Target

Some performance measures such as batch process metrics, derive the data for the calculation from objects within the system. The performance target can be linked to one or more related objects to define the specific data sources included in that target metric. For example, when creating a batch process performance target, the framework supports linking specific batch codes to the performance target to indicate the group of batch processes that are included in the measure.

The performance target business object may be configured to allow only relevant objects types to be linked to a performance target record. Refer to the base Batch Performance Target business object (`F1-BatchPerformanceTarget`) for an example.

## Creating Performance Target Zones

Once a performance target and its objects have been defined, a unique zone needs to be created to display and monitor the specific target.

The framework functionality for batch performance targets implements the zone creation via the related business object lifecycle. When a performance target is added, status enter plug-ins are responsible for generating the new zone using the template zone and prefix configured on the target category (or the override values) and adding the zone to the portal configured on the target type. When a performance target is inactivated, an enter plug in is responsible for removing the zone from the portal.

> **Note**
>
> While the template zone associated with the performance target category may be overridden, the zone generation algorithm makes certain assumptions about the zone type and parameters. In particular, the logic expects to configure a zone parameter that references the performance target code as input to the business service responsible for calculating the metrics.

Refer to the base Batch Performance Target business object (`F1-BatchPerformanceTarget`) for details of the lifecycle and the enter plug-ins responsible for performance target zone creation.

# Setting Up Performance Target Configuration

The following topics highlight the general configuration steps required to use performance target functionality. Your particular product or implementation may supply additional functionality to support specific use cases for performance targets. Refer to your product documentation and the library of business objects supplied for performance target in your system for more information.

- Performance Target Category Lookup
- Defining Performance Target Types

## Performance Target Category Lookup

Refer to About Performance Targets for an overview of performance target functionality.


Copyright © 2026, Oracle and/or its affiliates.

Each Performance Target Type is associated with a unique Performance Target Category. The categories are defined as an extendable lookup.

Navigate to the Extendable Lookup portal. Search for and select the **Performance Target Category** business object. Define values that are appropriate to the categories your implementation is assigning for the performance target types in use. In some cases, an edge application may already have delivered the appropriate performance target category for your use

Refer to the embedded help for more information about configuring this object.

## Defining Performance Target Types

Refer to About Performance Targets for an overview of performance target functionality.

To maintain the performance target types applicable to your product or implementation, open **Admin**, and then **Reports**, and then **Performance Target Type.**

This is a standard All-in-One portal.

The information captured on the performance target type depends on the business objects supported by your product or implementation. Refer to the embedded help text for more information.

# Maintaining Performance Targets

This section describes the functionality supported for viewing and maintaining performance targets.

Refer to About Performance Targets for an overview of performance target functionality.

Navigate using **Main**, and then **Reporting**, and then **Performance Target**. This is an All-in-One portal and includes the standard List and display zones for a performance target.

For information about the data defined on the performance target, refer to embedded help.

# Capturing Statistics

The product provides objects to allow for an edge product or an implementation to periodically calculate and capture statistics for one or more use cases. The topics in this section provide information about what is supplied in the framework and guidelines for implementing specific use cases.

- Understanding Statistics
- Configuring Your System for Statistics

## Understanding Statistics

There are two objects that work together to capture the statistics related to a given business use case:

- The statistics control record which defines configuration related to capturing statistics. It also defines a category that is used to group similar types of statistics together.
- The statistics snapshot is the record where the actual calculated statistics are captured.


Copyright © 2026, Oracle and/or its affiliates.

### Statistics Control

The statistics control record defines whether or not statistics are automatically calculated and how frequently. In addition, you may control the retention policy of snapshot records. You may indicate that only the most recent snapshot is kept or all snapshot records are kept or that they are retained for a defined number of days.

The system provides a business object for statistics control (`F1-Statistics`) that is expected to be used for most use cases. Its lifecycle includes the support for periodic capturing of statistics snapshot records as well as on demand capture.

### Statistics Snapshot

The statistics snapshot object is the object that is responsible for calculating and capturing the statistics details. A separate business object must be defined for each use case. The schema of the business object defines the details that are captured. In addition, it must include an algorithm as an enter plug-in on the **Complete** state that is responsible for capturing statistics.

The system provides a 'root' business object for statistics snapshot (`F1-SnapshotRoot`) which includes the lifecycle that statistics snapshot business objects should follow. No explicit statistics snapshot use case is provided by the Framework. Your specific product may supply some out of the box support for certain use cases, in which case specific statistics snapshot business objects are provided. Refer to your product specific documentation for more information.

### Viewing / Reporting Statistics

The product provides the mechanism for defining statistics control and statistics snapshot records and viewing the data in basic format. If your product has provided support for specific use cases, there may also be additional portals used to display the statistics in a meaningful format. For example, the statistics snapshot may be capturing data that can be presented in graphical format. Additionally, if multiple historical statistics snapshot records are retained, a zone may be defined to graph changes over time.

## Configuring Your System for Statistics

If your product provides statistics use cases that you are planning to implement, all that you need to do is configure the appropriate statistics control record and define the appropriate configuration for your business requirements. Refer to Defining and Monitoring Statistics for more information.

If your implementation has identified an additional use case where you would like to capture statistics, the following points highlight the steps needed to configure the system to support the use case.

- Determine whether an additional Statistics Category value is needed. This is captured on the statistics category record. Navigate to the Lookup page. Search for and select the `STAT_CATEGORY_FLG` field. Review the values and determine if additional values are needed.
- Define a new statistics snapshot BO. This should be a child BO of the base delivered BO (`F1-SnapshotRoot`). Its schema should define elements for the specific information that is captured by the statistics calculations. The schema should be designed in conjunction with an appropriate Enter Status algorithm for the **Complete** that calculates the statistics as appropriate for the business requirement.
- Once the snapshot business object is designed and implemented, configure the appropriate statistics control record.


Copyright © 2026, Oracle and/or its affiliates.

- Defining and Monitoring Statistics

## Defining and Monitoring Statistics

Refer to Understanding Statistics for an overview of the statistics functionality.

To define and maintain statistics control records and view a list of the current snapshot records, open **Admin**, and then **Reporting**, and then **Statistics Control.**

This is an All-in-One portal and includes the standard List and display zones for a statistics control.

For information about the data defined on the statistics control, refer to embedded help.

**Statistics Snapshot**

If there are any Statistics Snapshot records for the statistics control, the **Statistics Snapshot List** zone displays a list of the most recent records. A user may drill into any of the records to view more detail. You are brought to a maintenance portal where you may view details about the calculated statistics. The information captured will be unique to the particular use case. Refer to the embedded help for more information.

# Capturing Product Use Metrics

The product provides functionality to periodically calculate and capture information used to review and confirm licensing / subscription compliance for cloud implementations.

Following points highlight some of the features of this functionality:

- The product metric type object defines configuration related to a given metric, including which cloud offerings it applies to and the product metric snapshot business object to use when capturing the metric.
- The product metric snapshot object is used to capture the calculated metric.
- The Product Use Metric portal is provided to display information about the calculated metrics.
- A monitor batch control is used to create the product metric snapshots. Every time it's run, it creates a snapshot for each applicable product metric type. For on premises, all product metric types are considered applicable. For a cloud implementation, the system looks at the applicable cloud offerings setting on the product metric type. If that is populated with specific cloud offerings, then the product metric type is only applicable if your product is in the list. (If no applicable cloud offerings are populated, then it's considered applicable for all cloud offerings.)
- In a cloud environment, the monitor batch control used to create the product metric snapshot is considered an internal batch job and will be run by the Oracle Utilities Cloud Service.

> **Note**
>
> Although the product use metrics functionality is required for cloud implementations, implementations that are on premises may also choose to run the product metric type's monitor batch control to capture information for viewing in the Product Use Metric portal if desired.

- Defining Product Metric Type


Copyright © 2026, Oracle and/or its affiliates.

- Product Metric Snapshot
- Product Use Metrics

## Defining Product Metric Type

Product metric types are defined by the product. To view the product metric types, open **Admin**, and then **System**, and then **Product Metric Type.**

Use the **Product Metric Type Query** portal to search for a product metric type. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

For information about the data defined on the product metric type, refer to embedded help.

## Product Metric Snapshot

Product metric snapshots are created whenever the product metric type's monitor batch is run. A snapshot is created for every applicable product metric type. Refer to Capturing Product Use Metrics for more information

To view the product metric snapshots, open **Admin**, and then **System**, and then **Product Metric Snapshot.**

Use the **Product Metric Snapshot Query** portal to search for a product metric snapshot. Once a record is selected, you are brought to the maintenance portal to view the selected record.

For information about the data defined on the product metric snapshot, refer to embedded help.

## Product Use Metrics

The Product Use Metrics portal includes a zone (or zones) that are used to perform a business review of your organization's implementation of the product to confirm licensing / subscription compliance. The information visible on this portal will depend on the product or products installed by your organization.

To view this portal, open **Main**, and then **Dashboards**, and then **Product Use Metrics.**

For more information about what each zone on the portal is reporting, refer to the embedded help.

# External Messages

This section describes mechanisms provided in the product that enable an implementation to configure the system to communicate with an external application.

- Incoming Messages
- Outgoing Messages
- Web Service Category
- JMS Message Browser
- Oracle Integration Cloud Catalog


Copyright © 2026, Oracle and/or its affiliates.

- XAI Documentation Note

# Incoming Messages

This section provides information about support for incoming messages.

- Inbound Web Services
- Guaranteed Delivery

## Inbound Web Services

Inbound web service functionality is provided to support receiving web service requests from an external system.

- Understanding Inbound Web Services
- Configuring Inbound Web Service Options
- Deploying Inbound SOAP Web Services

### Understanding Inbound Web Services

The system supports communicating with the system via RESTful services or via SOAP services. In both cases, the system uses an object called inbound web service (IWS) to store the configuration. A web service class is used to distinguish whether the IWS is used for REST or SOAP. There is also a distinct business object for each web service class value because there is different configuration for each type of IWS.

For both types of inbound web service, the system supports the configuration of one or more operations per web service. Each operation defines the schema-based object to invoke to perform the desired function. An operation may refer to a Business Service, a Business Object, or a Service Script. If the IWS supports multiple operations, each operation can refer to the same or a completely different schema-based object from other operations within the IWS.

REST APIs may be secured by OAuth client credentials grant in a cloud installation. Refer to the Identity and Access Management chapter in the *Oracle Utilities Cloud Services Administration Guide* for more information.

The following topics provide more information about the different configuration of IWS based on whether it is SOAP or REST.

- Inbound REST Web Services
- Inbound SOAP Web Services

#### Inbound REST Web Services

This topic provides more information about the product's support of REST services using IWS.

The REST engine checks the Accept-Language header and attempts to find a supported language whose locale matches the referenced language. If found, the web service response returns any translatable data in that language. Note that if no supported language is found for the languages provided in the Accept-Language header, the system uses the language of the user that submitted the request.

IWS REST records are marked with a REST engine version of 1.0 or 2.0. Version 2.0 is the version used for new services. Version 1.0 is introduced for backward compatibility. Some of


Copyright © 2026, Oracle and/or its affiliates.

the system behavior differs based on the version of the REST service. The sections below highlight cases where this occurs.

### HTTP Method and Parameters

When defining operations for inbound REST web services, the product supports the HTTP methods of Get, Patch, Post, Put, and Delete. Note that the product's support of these various HTTP methods are a means of communicating the purpose of the web service to the outside world. However, the actual behavior of the REST web service is driven by the behavior of the underlying schema based object (business object, business service or service script). For example, you may configure the HTTP Method of "Put" for an operation that references a service script that just retrieves a record. The product is not able to detect this type of discrepancy. Configuration users should carefully consider the correct method to use based on the logic of that service.

For operations that reference a business object, the transaction type must be provided. The REST syntax doesn't support defining the transaction type at runtime. There are some basic validation checks in this case related to the transaction type and the HTTP method. For example, a Get method only makes sense with the Read transaction type.

> **Note**
>
> Using the transaction type **Change** requires all values to be passed in. Using the transaction type **Update** allows the web service to pass only the primary key and the values to be updated. All other elements will retain their existing values.

You may additionally define parameters. For each parameter, you define an external reference to the element, which is how this parameter is exposed to the external callers and is defined in the API specification. Each of these parameters is mapped to the XPath of the schema element from the underlying business object, business service or service script. For each parameter, you indicate if it is a **Path** parameter or a **Query** parameter.

- Path parameters are parameters that are part of the endpoint and are required. Each path parameter must be included in the operation's URI component surrounded by curly brackets.
- Query parameters are optional. They are not part of the endpoint but rather are included in the endpoint URL after a question mark, followed by name value pairs.

Refer to the URL section below for examples of path and query parameters in the sample URLs.

### URL Composition

When building the endpoint URL for a REST service, there are three main parts that make up the full URL.

- The first part is the one that is environment specific. This will differ for an on premise implementation as compared to a cloud implementation. Both will have the host and the port and then additional components that identify the environment.
- The second part is a hard coded path designated by the product, namely "/rest/apis".

  These two parts of the URL are defined in the substitution variable `F1_REST_BASE_URL`. This is defined when initializing your environment. Refer to the Server Administration Guide for more information.


Copyright © 2026, Oracle and/or its affiliates.

The remaining part of the URL is built dynamically based on configuration for each IWS and its operations. The components are "/ownerURIComponent/resourceCategoryURIComponent/iwsURIComponent/operationURIComponent"

- The owner URI component is taken from the owner flag of the inbound REST web service. A special extendable lookup (Owner Configuration for REST Services) defines this component for each owner flag.
- Each Inbound REST Web Service must reference a resource category. This category is used to associate related web services to a common category of resources. For multiple IWS records linked to the same resource category, external catalogues can use this information to group together related web services. The resource category is an extendable lookup and the URI component is an attribute of this record.
- Each REST IWS record defines a URI component that serves as an external identifier of this IWS record. The value must be unique within a given owner flag.
- Each operation must define the HTTP Method and optionally a URI component. When defining path parameters, the path parameters must be included in the URI component using curly braces.

In all cases, the URI component must begin with a slash ('/').

The following are some examples of the dynamic portion of the URL for a REST service. The last example illustrates the use of a query parameter.

| IWS Name URI Component | Owner URI Component | Resource Category URI Component | Operation HTTP Method URI Component | Dynamic URL Component | Run Time Example |
|---|---|---|---|---|---|
| c1rateCalculation / rateCalculation | C1 /customer | C1-RATES /rates | Post / | ../customer/rates/rateCalculation/ | ../customer/rates/rateCalculation/ |
| w1WorkActivity / workActivity | W1 /asset | W1-WORK /work | Get /{activityId} | ../asset/work/workActivity/{activityId} | ../asset/work/workActivity/5798165498 |
| w1WorkActivity / workActivity | W1 /asset | W1-WORK /work | Patch /scheduleWindow/{externalSystem}/{activityId}/{windowStartDateTime} | ../asset/work/workActivity/scheduleWindow/{externalSystem}/{activityId}/{windowStartDateTime} | ../asset/work/workActivity/scheduleWindow/MY-COMPANY/5798165498/20190101 |
| CM-AccountActivityHistory / accountActivityHistory | CM /cm | CM-ACCT-INFO / accountInformation | Get /{accountId} | ../cm/accountInformation/accountActivityHistory/{accountId} | ../cm/accountInformation/accountActivityHistory/123456789?activityId=5468976 |


Copyright © 2026, Oracle and/or its affiliates.

### Payload Format

REST services support receiving a request payload in XML or JSON format and returns the payload in either XML or JSON. The default format returned depends on the value of the REST Engine Version.

- Version 2.0 services assume JSON format as the default. The default can be overridden by providing an accept header of `application/XML`.
- Version 1.0 services assume XML format as the default. The default can be overridden by providing an accept header of `application/JSON`.

### Root Node for JSON Format

Version 2.0 services with the JSON formatted payload do not accept in the request or return in the response any root node around the payload. The following is an example of the response of a REST call for a version 2.0 service:

```
{
   "batchJobId": "string",
   "requestSuccessful": "string"
}
```

Version 1.0 services with the JSON formatted payload expect a root node in the request and return one in the response. The following is an example of the response of a REST call for a version 1.0 service:

```
{
  "F1CnclBatJob": {
    "batchJobId": "string",
    "requestSuccessful": "string"
 }
}
```

The Open API Specification visible on the inbound web service maintenance page displays the expected format based on the record's REST engine version when viewing the specification.

### Data Type Format in JSON

In JSON format, strings are surrounded by quotes and numbers and Boolean data do not have quotes. All services for REST engine version 2.0 follow this standard. Originally services that are REST engine version 1.0 were incorrectly treating numbers and Boolean data as strings and returning the data with quotes. This has been corrected.

To accommodate any integration that may have worked around the behavior for version 1.0 services, the system provides the ability to identify inbound web services that are exceptions to this rule. For any inbound web service identified as an exception, the system continues to treat numbers and Boolean data as strings in JSON responses. To add one or more IWS records to the exception list:

- Go to Feature Configuration.
- Look for an existing feature configuration with the feature type **External Messages**. If one does not exist, create one.
- Add an option for the option type **IWS JSON Data Type Exceptions**.


Copyright © 2026, Oracle and/or its affiliates.

- In the option value, indicate the IWS record that is an exception. Note that multiple options for the option type may be added. Additionally, the option value supports a comma delimited list.

### External Facing Schema

By default the schema of the underlying service (for example, service script, business object or business service) is also the schema for the REST inbound web service operation and acts as both its Request and the Response schema. The product provides the ability to define an explicit schema for a REST inbound web service (IWS) operation, allowing a user to adjust the schema for an external facing consumer. The IWS operation schema also supports some special configuration that allows additional features to be defined only for the IWS operation schema.

The operation schema supports the following capabilities:

- Declaring whether an element belongs only to the request schema, only to the response schema, both (default) or excluded altogether. This is used to establish a clear separation between request and response schema definitions.
- Assign a different element name to an internal element. This allows the internal service element name to more closely align to internal references to the element, if needed. The external facing element name can be different allowing for a more readable schema.
- Introduce specialized elements that support links and other types of structural standards in a way that does not impact internal service processing.

Refer to Web Service Schema Nodes and Attributes for more information.

The REST servlet uses the request and response schemas to map elements to and from the internal schema.

> **Note**
>
> While the caller should only provide elements defined by the request schema, the application does not filter out extraneous elements. Assumption is that the internal service is designed to ignore such elements. However, only elements defined by the response schema are included in the response.

### Dynamic Links

There are use cases where the published API includes a "_self" element that contain the endpoint URL of the GET operation related to the data returned in the response. Additionally, response payloads may include foreign keys and for those entities, the response includes a "_link" element that contains the endpoint URL of the GET operation for that entity (if it exists).

Syntax is provided in the REST inbound web service operation schema to support building the runtime endpoint URL for the _self and _link elements. Besides dynamically building the static portion of the URL based on the current environment details, it also builds the dynamic portion of the URL, substituting the URL components for the operation and substituting the path parameters. The syntax allows you to define a specific IWS Operation or allows you to reference a maintenance object and at runtime, the system determines the GET operation for the maintenance object and builds the URL accordingly.

Example of the syntax:

```
<_link getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/>
```


Copyright © 2026, Oracle and/or its affiliates.

Example of the runtime endpoint URL:

```
_link: "http://.../common/toDos/toDoEntries/28937296450934"
```

A GET operation may be associated with a maintenance object to denote that this is the default GET operation for the entity. A **GET Operation** option at the Maintenance Object and Business Object level may be used to override this default. If the entity is not associated with a business object or the latter is not associated with such option then the operation associated with the maintenance object (if any) is used.

### API Specification Documentation

The following documentation information may be provided for an Inbound Web Service:

- Short and detailed description for each operation.
- Help text may be provided for individual elements. The same text can be shared across multiple elements across all operations of the Inbound Web Service. If an element is not associated with help text then its internal field's label is used instead.
- Sample request and response documents may be provided for each operation.
- Operations may be associated with a sequence number that controls the order in which they appear in the Open API Specification.

> **Note**
>
> Detailed descriptions and element related help text are not translated since the published catalogue for base product APIs is in English language only.

### Customization Options

Some base product REST web services may explicitly indicate that they allow customization. When allowed, your implementation may customize the following aspects of a service to meet integration requirements:

- Add more elements to the request and response schemas
- Implement custom logic to populate additional elements or enforce custom business rules.
- Provide help text for the additional elements and corresponding request and response examples when generating an API specification.

When an IWS allows customization, a **Custom Script** may be specified on any of its IWS operations to override the operation's underlying base service.

> **Note**
>
> The custom script must adhere to the base service schema it overrides and can only extend it. For that reason, the custom script's schema must at least include the operation's base service schema and may include additional custom elements as needed.


Copyright © 2026, Oracle and/or its affiliates.

Customizing an IWS operation does not change the URL used to access the web service. If the operation is customized then internally the custom script is called instead of the base service.

In the same way, the API specification of the operation is adjusted to use the custom script if any. If a custom web service schema is also provided then it is used instead of the base web service schema if any. If sample request and response sample documents are provided as options on the customized operation they are used instead of base samples. You may also provide custom help text for your custom elements by adding corresponding entries to the IWS help text list.

## Inbound SOAP Web Services

For inbound SOAP web services, by default the system uses the Schema Name to dictate the Request and Response for the service. The API can be overridden with custom formats by specifying Request and Response XSLs to transform into the relevant schema formats. In addition, if desired, the Request and Response Schemas that document the expected message may be provided.

> **Note**
>
> The Request and Response Schema fields are not supported in a cloud implementation.

> **Note**
>
> The system supports defining the XSLs as a managed content record. For backward compatibility, the system supports defining the XSL as a file in the file system. This is a system wide setting, defined using a feature configuration option. The feature type is **External Messages** and the option type is **XSL Location**. Set the value to `F1FL` to support XSL in the file system. The file location is configured using a system property. To support the XSL in managed content, no option is needed as this is the default. You may explicitly define the value of `F1MC` if desired.

In addition, note that for business object based operations, when invoking the web service an action is required. This may be passed into the web service as part of the invocation or alternatively, the action may be defined when configuring the operation using the transaction type.

> **Note**
>
> Using the transaction type **Change** requires all values to be passed in. Using the transaction type **Update** allows the web service to pass only the primary key and the values to be updated. All other elements will retain their existing values.

### Annotations Used for Security

When preparing to deploy inbound SOAP web services, the security aspects of the service must be decided. The product provides a default security policy that is applied when no other policy is defined: **@Policy(uri="policy:Wssp1.2-2007-Https-BasicAuth.xml", attachToWsdl=true)** which requires HTTP Basic over SSL and a WS-Security Timestamp.


Copyright © 2026, Oracle and/or its affiliates.

If a different security policy is desired, the following options are available:

- Security policies may be attached to the Inbound Web Service via the Java Enterprise Edition (Java EE) Web Application Server. This allows for multiple policies to be attached as supported by the Java EE Web Application Server. In order to enable this capability, explicit system configuration is required so that the product does not assume the default security policy. See the subsequent bullets for more information.
- Define a system wide security policy using a feature configuration option. Find the Feature Configuration record for the **External Messages** feature type. (It may need to be defined if it does not exist). Choose the option type **Default security policy** and define an appropriate value. If your implementation wishes for the policies to be attached at the Java EE Web Application Server, define this option type with an option value of `<none>`.
- Attach a security policy to the IWS via a Web Service Annotation. The base product provides annotation types that support the standard WS-Policy (`F1POLICY`) and OWSM Security Policy (`F1-OWSM`). No base annotation is supplied by the product for either annotation type.

  If your implementation wishes for the policy of a particular IWS to be attached at the Java EE Web Application Server, define a special annotation for the `F1POLICY` annotation type and configure the **uri** parameter value to `<none>`.

> **Note**
>
> Refer to WebLogic documentation for more information on supported security policies.

> **Note**
>
> In order to use the OWSM Policy, additional system configuration is necessary. Contact your system administrator to confirm if your implementation supports OWSM.

**Inbound SOAP Web Service Deployment**

A Inbound SOAP Web Service must be deployed to the Java EE Web Application Server in order for it to be available to the Web Service Clients to access the system. Refer to Deploying SOAP Web Services for more information.

- Deploying XAI Inbound Service via IWS

## Deploying XAI Inbound Service via IWS

For implementations using XAI inbound services for external messages, the product recommends moving to the inbound web service mechanism, which uses the Java EE Web Application Server to communicate with the product rather than the XAI servlet.

For XAI inbound services that use the **Business Adapter**, it is straight forward to move to IWS because the configuration is similar. In both cases, the service is configured to reference a business object, business service or service script. The associated WSDL for each record is similar. Changing the interface for the incoming message to use IWS instead of XAI inbound services is similar.

However, for XAI inbound services that use the **Core Adapter**, these services reference an underlying "page service" in the product. For these services, the Request and Response schemas for the XAI inbound service were created using the Schema Editor. In order to support calling an underlying "page service" in IWS, first a business service must be created to


Copyright © 2026, Oracle and/or its affiliates.

reference the page service (if one doesn't already exist). However, the resulting schema for the business service is different from the Request and Response schemas related to the XAI inbound service. Moving this functionality to IWS using business services requires changes to the format of the incoming messages.

Moving all incoming messages over to use IWS instead of XAI is the product recommendation. However, to aid in implementations that have many integrations in place using the XAI inbound services that use the **Core Adapter** (or any adapter whose message class is **BASEADA**), the product provides the ability to deploy these types of XAI inbound services to the Java EE Web Application Server along with the Inbound Web Services.

To take advantage of this capability, you must define a feature configuration option. Under the **External Messages** feature configuration type, the **Support XAI Services via IWS** is used to indicate if this feature is supported. Setting the value to **true** turns on the feature. If no option is defined for that option type, it is equivalent to setting the value to **false**.

When the system is configured to support XAI services via IWS, the Inbound SOAP Web Service deployment includes XAI inbound services (that are configured with an Adapter that references the **BASEADA** message class). The deployment portal will also include a zone showing the deployment status of these XAI Inbound Services.

> **Note**
>
> There is no support for XAI inbound services via REST, only via inbound SOAP web services.

# Configuring Inbound Web Service Options

This topic describes the configuration needed for using inbound web services.

- Configuring SOAP Inbound Web Service Options
- Configuring REST Inbound Web Service Options
- Maintaining Inbound Web Services
- Maintaining Inbound Web Service Operations

## Configuring SOAP Inbound Web Service Options

This topics in this section describes the configuration needed for using inbound SOAP web services.

- Technical Configuration
- Maintaining Web Service Annotation Types
- Maintaining Web Service Annotations
- Maintaining XAI Inbound Services

### Technical Configuration

In order to use inbound web services, there are tasks a system administrator must perform.

Refer to the Server Administration Guide for technical details of each of these processes.


Copyright © 2026, Oracle and/or its affiliates.

## Maintaining Web Service Annotation Types

The product provides some base annotation types. Refer to the metadata for more information. If your implementation wishes to define additional annotation types, use the Web Service Annotation Type portal. Open this page using **Admin**, and then **External Message**, and then **Web Service Annotation Type.**

You are taken to the query portal where you can search for an existing web service annotation type. Once an annotation type is selected, you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> Use of custom policies should only be considered if the policies supplied by the Java EE Web Application Server are not sufficient for your implementation's needs.

> **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the web service annotation type code. Refer to System Data Naming Convention for more information.

The **Web Service Annotation Type** zone provides basic information about the web service annotation type.

Please see the zone's help text for information about this zone's fields.

The system supports the ability for an IWS record to refer to multiple policies. In this situation, the annotation type for the policy should include a reference to a parent annotation type so that the system can properly build the array of annotations.

## Maintaining Web Service Annotations

If your implementation wishes to define annotations, use the Web Service Annotation portal. Open this page using **Admin**, and then **Integration**, and then **Web Service Annotation.**

This is a standard All-in-One portal.

> **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the web service annotation code. Refer to System Data Naming Convention for more information.

## Maintaining XAI Inbound Services

XAI Inbound Service is a legacy mechanism for exposing system services to external systems as web services. The current supported mechanism for defining web services is Inbound Web Services, which should be used for any new web service.


Copyright © 2026, Oracle and/or its affiliates.

Some framework based products and existing implementations may have existing XAI inbound services that may need to be viewed. For these services, the product supports deploying them as SOAP inbound web services for execution. Refer to Deploying XAI Inbound Service via IWS.

The following sections describe basic information about the maintenance pages. Note that some of the information is related to legacy functionality that is no longer supported.

### XAI Inbound Service - Main

To view an inbound service, open **Admin**, and then **XAI**, and then **XAI Inbound Service**.

#### Description of Page

**XAI In Service Name** is used in the system to identify the service. The service name is also the first XML element after the <Body> element in the XML request/response document. The **XAI Service ID** serves as the primary key.

**Owner** indicates if this XAI inbound service is owned by the base package or by your implementation ( **Customer Modification**). This information is display-only.

The **Adapter** defines the interface with the target application server.

If adapter for this service invokes a system service, then the appropriate **Service Name** is visible.

If adapter is the **Business Adapter** then **Schema Type** and **Schema Name** reference the object to invoke.

**Web Service Category** is visible if the XAI inbound service is linked to one or more web service categories.

Use the **Description** and **Long Description** to describe the service.

Check the **Active** switch if this service is enabled and available for execution.

**Post Error** is not applicable.

**Trace** is not applicable.

**Debug** is not applicable.

#### Schema Definitions

> **Note**
>
> Request Schema and Response Schema are not applicable to services invoking schema-based objects. They do not appear when the **Business Adapter** is used.

The next two properties define the request and response XML schemas. The schemas are SOAP compatible. The schema XML files are expected to be stored in the Schemas Directory on the Web server running the XAI server.

The **Request Schema** is the XML schema defining the service request. The request sent to the server must adhere to the schema definition.

The **Response Schema** is the XML schema defining the service response. The response generated by the XAI server corresponds to the response schema definition.


Copyright © 2026, Oracle and/or its affiliates.

The same service may perform several actions on a business object. Use the **Transaction Type** to define the default action performed by a service. The transaction type can be provided when invoking a service, by dynamically specifying a transaction type attribute on the Service element of the XML request. This field may take the following values: **Read**, **Add**, **Change**, **Update**, **Delete**, **List** and **Search**.

> **Note**
>
> The difference between **Change** and **Update** is that for **Change**, all field values must be passed in with the request. Field values that are not passed in to the request are set to null. For **Update**, you need only pass the primary key field values and the values of the fields to be updated. All other fields retain their existing values.

Services, which perform a Search, may allow searching based on different criteria. When the Transaction Type value is **Search**, use the **Search Type** to define the default search criteria. The possible values are **Main**, `Alternate1`, `Alternate2`, `Alternate3`, `Alternate4`, `Alternate5` and `Alternate6`.

> **Note**
>
> This is a default definition only and it may be overridden at run time when the service is invoked. To override the search type at run time, you should specify the searchType attribute on the Service element of the XML request.

#### XSL Transformation Definitions

Sometimes, the XML request document does not conform to the request schema, or the response document expected by the service requestor is not the one generated by the adapter. In such cases the request and/or the response documents must be transformed. The XAI server supports transformation through XSL transformation scripts. Transformation scripts may be applied to the request before it is passed to the adapter or applied to the response document before it is sent to the service requestor.

The **Request XSL** is the name of the XSL transformation to be applied to the request document before processing it. The transformation is usually required when the incoming document does not correspond to the XAI service request schema therefore it has to be transformed before it can be processed by the adapter.

The **Response XSL** is the name of the XSL transformation to be applied to the response document when the requester of the service expects the response to have a different XML document structure than the one defined by the response schema for the service.

Click the **WSDL URL** hyperlink to launch a separate window that contains the WSDL definition for the inbound service. Note that the server name and port number for the URL are built using a setting in the common properties file using the XAI HTTP Caller URL setting.

## Configuring REST Inbound Web Service Options

This topics in this section describes the configuration needed for using inbound REST web services.

- Owner Configuration Lookup
- Resource Category Lookup


Copyright © 2026, Oracle and/or its affiliates.

## Owner Configuration Lookup

Refer to Inbound REST Web Service for an overview of REST IWS functionality.

When generating the URL for a REST web service, part of the URL is taken from the owner of the Inbound REST Web Service record. The component to add to the URL is defined in this lookup using the URI Component field. The values are provided by the product.

To view the values, navigate to the Extendable Lookup portal. Search for and select the **Owner Configuration for REST Services** business object.

## Resource Category Lookup

Refer to Inbound REST Web Service for an overview of REST IWS functionality.

Each Inbound REST Web Service is associated with a Resource Category. The categories are defined as an extendable lookup. The REST URL is built using a component from the IWS record's resource category.

Navigate to the Extendable Lookup portal. Search for and select the **Resource Category** business object. Define values as needed. Typically the category is related to a logical grouping of entities, or maintenance objects, referred to as a 'resource' in REST nomenclature. In some cases, an edge application may have already delivered appropriate resource categories for your use.

Populate the appropriate URI component to include when building the URL for IWS records linked to this resource category.

# Maintaining Inbound Web Services

Inbound Web Services are used to define a specific message that your implementation will receive from an external system and provides configuration needed to process the inbound message.

The product provides support for REST and SOAP services and provides several inbound web services out of the box.

For inbound SOAP web services, by default no annotations are defined for the base inbound web services. You may modify the message options or the annotations for any base IWS record. In addition, you may define additional IWS records for other incoming messages supported by your implementation.

To view an inbound web service, navigate using **Admin**, and then **Integration**, and then **Inbound Web Service.** You are brought to a query portal with options for searching for inbound web services.

Once an inbound web service has been selected, you are brought to the maintenance portal to view and maintain the selected record.

> ⚠ **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the inbound web service code. Refer to System Data Naming Convention for more information.

The following zones may appear as part of the portal's **Main** tab page


Copyright © 2026, Oracle and/or its affiliates.

- **Inbound Web Service.** This zone displays the configuration information for the record and its operations.
- **Operations.** This zone lists the operations defined for the selected inbound web service. The zone is only visible for a REST inbound web service. You may navigate to the Inbound Web Service Operation portal to further describe the operation's web service schema.
- **Help Text Fields.** This zone maintains help text information for fields across all operations of the selected inbound web service. The zone is only visible for a REST inbound web service.

Refer to Inbound SOAP Web Services for specific information about Operations for SOAP records.

Refer to Inbound REST Web Services for specific information about Operations for REST records.

Note that in addition to standard actions available on this portal, there is also a special **Add to Category** button in the page action area. Click this button to link an inbound web service to one or more web service categories.

## Maintaining Inbound Web Service Operations

This portal is used to maintain a REST inbound web service operation. You can access the portal from the **Operations** zone on the Inbound Web Service portal for a REST service.

Refer to Inbound REST Web Services for specific information about Operations for REST records.

The following zones may appear as part of the portal's **Main** tab page

- **IWS Operation.** This zone provides information about the selected operation.
- **Schema.** This zone describes the request and response schema as defined by the operation's web service schema. The zone is only visible when an operation schema is explicitly defined.
- **Schema Extensions.** This zone manages web service schema extensions that correspond to extensions made to the operation's internal schema. This zone is only visible if the underlying schema was extended using dynamic data area extension functionality. Each owner that extended the internal schema may provide web service schema information for their extension elements.
- **Options.** This zone lists various options for the operation such as request and response samples. The zone is visible when such options exist for the operation. Note that options are maintained as part of operation record.

The following zones may appear as part of the portal's **References** tab page

- **IWS Operation References.** This zone lists references to the current web service operation. The list includes other inbound web service operation records that reference the current operation via their web service schema. For a GET operation, the list also includes the maintenance object or business object that explicitly references the operation using a **GET Operation** option.
- Web Service Schema Nodes and Attributes

## Web Service Schema Nodes and Attributes

A REST inbound web service (IWS) operation invokes a single schema based object, either a Business Object, a Business Service or a Service Script. The object has one schema which by default acts as both the Request and the Response schema. The product provides the ability


Copyright © 2026, Oracle and/or its affiliates.

to define an explicit schema for a web service operation that represents a view of the underlying internal schema for an external facing consumer. Note that the operation schema is a strict view of the internal schema and as such does not introduce structural changes.

While optional, defining an operation schema has the following benefits:

- Declaring the **usage** of an element as to whether it is part of the Request schema, Response schema, both or neither.
- Assigning an **external element name** to an internal element. Useful to avoid cloning of internal services for this purpose only.
- Better management and visibility of **GET operation** references.
- Support HATEOAS style structures outside of the internal schema. This allows for seamless internal processing.
- Providing help text for individual elements.

The following documentation is a complete list of the attributes available to you when constructing an IWS operation schema.

### Operation Schema Definition

The following attributes defines the external facing operation schema that establishes the request and response schema for the operation.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| usage= | "REQ" | Indicates that the element should be included in the request schema only.<br>When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default. | `<customerName usage="REQ" />` |
| | "RESP" | Indicates that the element should be included in the response schema only.<br>When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default. | `<customerId usage="RESP" />` |
| | "BOTH" | Indicates that the element should be included in both request and response schema definitions. This is the default value<br>When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default. | `<customerName usage="BOTH" />` |
| | "EXCL" | Indicates that the element should not be included in either the request or the response schema definitions.<br>When specified for a group or list element the value applies to the entire container. A specific element in the container may override the container default. | `<httpMethod usage="EXCL" />` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| mapTo= | "internal element name" | Maps a web service element to an internal element. By default the internal element name is assumed to be the same as the web service element name. Use this attribute to assign an external facing element name in the operation schema and map it to a corresponding internal element.<br><br>Note that this is not an XPath expression but rather a reference to a valid element name that corresponds to the structure of the expanded internal schema.<br><br>The owner of the element must be the same as the owner of the web service schema. If an operation's underlying schema was extended (using dynamic data area extension) then the extending owner may only reference their elements on their web service schema extension for the operation. | Internal schema element:<br><br>`<toDoPriority .../>`<br><br>Operation schema element:<br><br>`<priority mapTo="toDoPriority"/>` |
| role= | "FKGP" | Establishes a group structure that represents a foreign key to an entity.<br><br>The element itself does not exist in the internal schema. It groups together the foreign key value element(s) along with a corresponding `_link` element that at run time provides the endpoint URL of that entity's GET operation. Consumers of the web service may use the link to get the data for the entity. | Internal schema element:<br><br>`<user .../>`<br><br>Operation schema group:<br><br>`<user role="FKGP">`<br>`  <user>`<br>`  <_link getOperation="mo:'USER';pk1:user;"/>`<br>`</user>` |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | "COLL" | Establishes a group structure that represents a collection of entities that corresponds to an internal service list.<br>The element itself does not exist in the internal schema. It groups together the internal list mapped to a _data element along with a corresponding _link element that at run time provides the endpoint URL for the collection's GET operation. Consumers of the web service may use the link to get a more detailed set of information for each entity in the list.<br>By default all the elements in the referenced list are included in the operation schema. You may explicitly reference and define attributes for specific elements as needed. | Internal schema element:<br><alerts type="list"><br><alert/><br><startDate /><br><endDate /><br><version /><br></_data> </alerts><br>Operation schema group:<br><alerts role="COLL"><br><_link getOperation="..."/><br><_data mapTo="alerts"><br><version usage="EXCL"/><br></_data><br></alerts> |
| getOperation= | "expression"<br>Expression components:<br>• mo:'<maintenance object>';<br>• pk1:<relative xpath to an element in the internal schema>;<br>• pk2:...;<br>• pk3:...;<br>• pk4:...;<br>• pk5:...;<br>Parameters pk2–5 are required to match the maintenance object's prime key definition. | Only valid on _self and _link elements. The attribute is used to reference the default GET operation for an entity as defined by its maintenance object or business object.<br>The application determines the GET operation to use as follows:<br>• If the entity's business object references an operation using the **GET Operation** option then it is used.<br>• If the entity's maintenance object references an operation using the **GET Operation** option then it is used.<br>• If a GET operation is found that references the maintenance object then it is used.<br>If a GET operation cannot be determined for the entity a standard text indicating that a link is not available is returned instead. | <_link getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/> |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| | “expression”<br>Expression components:<br>• `iws:`’<inbound web service name>’;<br>• `operation:`’<operation name>’;<br>• `parms:` [<external reference as defined on the GET operation record>:<relative xpath of an element name in the internal schema that holds the value>; <another external reference name>:<another relative xpath>;...]; | Only valid on _self and _link elements. The attribute is used to reference a specific GET operation along with the information needed to compose its path parameters. | `<_self getOperation="iws:'F1-ToDoEntry'; operation:'getToDoEntry'; parms:[toDoEntryId:toDoEntryId;]"/>` |

### Specialized Elements

The operation schema definition makes use of specialized element names in the style of the Hypermedia as the Engine of Application State (HATEOAS) standard. These elements should only be defined by the operation schema as they are not relevant to nor should be managed by the internal operation service.

The following list describes each element and how it is used.

| Element Name | Description | Examples |
|---|---|---|
| `<_self ... />` | Response payloads may include a "_self" element that contains the endpoint URL of the GET operation related to the entity returned in the response.<br>This is an optional element and only one may exist in an operation schema.<br>Refer to `getOperation=` schema definition attributes for more information. | `<_self getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/>` |
| `<_link ... />` | Response payloads may include foreign keys and for those entities, the response includes a "_link" element that contains the endpoint URL of the GET operation for that entity (if it exists).<br>Refer to `role=` and `getOperation=` schema definition attribute for more information. | `<_link getOperation="mo:'TO DO ENTRY';pk1:toDoEntryId;"/>` |


Copyright © 2026, Oracle and/or its affiliates.

| Element Name | Description | Examples |
|---|---|---|
| <_data ... /> | Provides a standard reference to a list of entities in the internal schema.<br>Refer to role="COLL" schema definition attribute for more information. | <alerts role="COLL"><br><_link getOperation="... "/><br><_data mapTo="alerts"><br></_data><br></alerts> |

### Documentation Attributes

The following attributes provide additional documentation for individual elements that is included as part of the web service's Open API specification.

| Mnemonic | Valid Values | Description | Examples |
|---|---|---|---|
| helpText= | "help text field name" | Associates an element with a specific help text record defined for the Inbound Web Service. If the element is included in both request and response schema the same help text is used in both.<br>If this attribute is not specified, and a single Help Text Field record exists for the Field associated with the internal element, that help text record is used. If more than one such Help Text Field record exists the one where the help text name is the same as the Field name is used.<br>Note that an internal element is associated with a Field name via the mdField= attributes, if any, else via the mapField= attribute.<br>No valid for _self and _link elements. | <toDoEntryId usage="BOTH" helpText="TD_ENTRY"/><br><br>In this example the inbound web service defines a help text field name "TD_ENTRY" that provides help text for this element. The same help text is included for the element in both request and response schema. |
| helpTextResponse= | "help text field name" | Associates an element that is included in both request and response schema with a specific help text record to be used in the context of the response schema.<br>Follows same rules as the helpText= attribute. | <toDoEntryId usage="BOTH" helpText="TD_ENTRY" helpTextResponse="TD_ENTRY_RESP"/><br><br>In this example the inbound web service defines the following help text records:<br>• Help text field name "TD_ENTRY" describes the element sent in the request schema.<br>• Help text field name "TD_ENTRY_RESP" describes the element populated in the response schema. |


Copyright © 2026, Oracle and/or its affiliates.

# Deploying Inbound SOAP Web Services

Once an Inbound SOAP Web Service is defined it is not automatically available to the Web Service Clients to access the system. The Deployment Status and the Active flag (set to true) indicate whether a Web Service is available or not. The last step is to deploy the Inbound SOAP Web Services to the Java EE Web Application Server. This deployment phase has a number of steps that are automatically performed when a deployment is initiated:

- The Web Service files are generated and policies are attached.
- The WSDL is generated with appropriate annotations and enumerations.
- The necessary Java stub code to implement the Web Service in the Java EE Web Application Server is generated and compiled.
- The Web Services are built into a valid Web Application Archive (WAR) file.
- Optionally, the newly created Web Services WAR file is deployed to the Java EE Web Application Server. This can also be done manually for clustered deployments, if desired.

There are two methods available for deploying inbound SOAP web services:

- Deployment at the command line using the **iwsdeploy[.sh]** command as outlined in the Server Administration Guide. This method is recommended for native installations and production implementations.
- Deployment using the Inbound Web Service Deployment portal. This method is only supported in development (non-production) environments.

### Inbound Web Service Deployment Portal

To use the online Inbound Web Service Deployment portal, navigate using **Admin**, and then **Integration**, and then **Inbound Web Service Deployment**.

The following sections describe the base zones that are provided on the portal.

### Deploy Inbound SOAP Web Services

The Deploy Inbound SOAP Web Services zone provides information about the last deployment. If the region is a development (non-production) region you may use the **Deploy** button to deploy or re-deploy inbound web services. All inbound web services whose Active switch is Yes will be deployed. All whose active switch is No will be undeployed.

> **Note**
>
> When an Inbound SOAP Web Service is deployed, the value of its service revision field is captured. Certain changes to configuration will require re-deployment to take effect. When any of the following changes occur, the IWS service revision value is incremented. This will cause the deployment status to show **Needs Deployment**.
>
> - Active switch is changed
> - An Annotation is added or removed
> - An Operation is added or removed.
> - The Operation Name, Schema Type / Schema Name, Request or Response Schema, Request or Response XSL for an Operation is changed.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In addition, if the implementation supports XAI services deployed through IWS, the appropriate XAI inbound services will also be deployed or undeployed as required.

**Deployment Status**

The Deployment Status zone displays a list of inbound SOAP web services in the product, including the deployment status.

The deployment status is determined by comparing the internal Service Revision field on each IWS against the value captured at the time of deployment.

- **Deployed**. Indicates that the IWS has been deployed and no changes have been detected to the configuration.
- **Needs Deployment**. Indicates that the IWS has never been deployed or has been deployed but in the meantime, changes have been detected to the configuration that require redeployment.
- **Undeployed**. Indicates that the IWS is marked as inactive and the IWS is not found to be deployed at this time.
- **Needs Undeployment**. Indicates that the IWS is marked as inactive but the IWS is found to be deployed at this time.

If the IWS has been deployed, the View column will include a **WSDL** link allowing you to launch a separate window to view the WSDL definition.

> **Note**
>
> For cloud installations, the system supports defining the WSDL URI via a substitution value defined for the token URI: @F1_BASE_IWS_URI@. The system will use this value per the functionality described in URI Substitution. When no value is defined, the system uses a default URL, which is the appropriate URL for on-premise installations.

Use the broadcast button adjacent to any of the inbound web services listed in the zone to view the details of the IWS record. This causes the **Inbound Web Service** zone to appear. It is the same zone that appears on the Inbound Web Service maintenance portal.

**XAI Inbound Service Deployment Status**

The XAI Inbound Service Deployment Status zone is only visible if the feature configuration option **Support XAI Services via IWS** is configured on the **External Messages** feature type or if the system detects that there are XAI inbound services that have been deployed. (The latter condition is checked for the case where an implementation has XAI inbound services deployed and then chooses to discontinue using this functionality. After changing the feature configuration option to false, one more deployment is required to "undeploy" the XAI services.) The zone displays a list of XAI inbound services in the product that are related to page services. Refer to Deploying XAI Inbound Service via IWS for more information.

The deployment status is determined by comparing the record's Version field against the value captured at the time of deployment.

- **Deployed**. Indicates that the XAI inbound service has been deployed and no changes have been detected to the configuration.


Copyright © 2026, Oracle and/or its affiliates.

- **Needs Deployment**. Indicates that the XAI inbound service has not been deployed or has been deployed but in the meantime, changes have been detected to the configuration.
- **Undeployed**. Indicates that the XAI inbound service is marked as inactive or the **Support XAI Services via IWS** is not set to **true** and the XAI inbound service is not found to be deployed at this time.
- **Needs Undeployment**. Indicates that the XAI inbound service is marked as inactive or the **Support XAI Services via IWS** is not set to **true** but the XAI inbound service is found to be deployed at this time.

XAI inbound service does not have the equivalent of a Service Revision field that inbound web service has, which is only incremented when changes are made to the record that impact deployment. For XAI inbound service, the version number on the record is used. This field is incremented when any changes are made, even ones that may not impact deployment. As a result, some XAI Inbound Services may indicate "Needs Deployment" in cases where a redeployment may not be necessary. The recommendation when this occurs is to simply Deploy again to be safe.

If the IWS has been deployed, the View column will include a **WSDL** link allowing you to launch a separate window to view the WSDL definition.

## Guaranteed Delivery

There are alternatives for sending messages to the system besides using inbound web services. An external system may be able to send messages to the system in a generic manner where a new web service does not need to be defined for every new type of message. These types of messages may provide a payload (the message) and the service script or business service to invoke. An example of this type of communication is a message sent from a mobile application using RESTful operations.

The external system may have no mechanism for retrying failed messages. For this situation, the product provides an algorithm that may be used to capture incoming messages that should "guarantee delivery". A servlet processing this type of message may invoke the installation algorithm - Guaranteed Delivery, passing the details of the message and an indication if a response should be returned. The algorithm is responsible for storing the message information in a table so that it can be subsequently processed.

# Outgoing Messages

"Outgoing messages" is the term used to describe messages that are initiated by our system and sent to an external system. Messages may be sent real time or near real time. The system provides the following mechanisms for communicating messages to external systems.

- **Outbound Messages.** This method allows implementers to use configurable business objects to define the message format and to use scripts to build the message. If sent near real-time the message is posted to the outbound message table waiting for Oracle Service Bus to poll the records, apply the XSL and route the message. If sent real time, the message dispatcher routes the message immediately.
- **Web Service Adapters.** Using a web service adapter, an implementation can consume a WSDL from an external system and create an "adapter" record that references the URL of the external system and creates appropriate request and response data areas to expose the payload information in a format understood by configuration tools. A script may then be written to build the request information and initiate a real-time web service call from within the system.


Copyright © 2026, Oracle and/or its affiliates.

- **Send Email.** The system supplies a dedicated business service that may be used to send an email real-time from within the application.

All these methods are described in more detail in the following sections.

- Outbound Messages
- Web Service Adapters
- Sending Email

# Outbound Messages

Outbound messages provide functionality for routing XML messages to an external system real-time or in near real time. In addition the functionality supports collecting related messages into a batch to then be sent to an external system as a consolidate XML message.

For each outbound message that your implementation must initiate you define a business object for the outbound message maintenance object. Using the business object's schema definition, you define the fields that make up the XML source field. These are the fields that make up the basis for the XML message (prior to any XSL transformation).

Each outbound message requires the definition of its schema by creating a business object whose schema describes the information that is provided to the external system. An XSL transformation may then be performed when routing the message to an external system.

For each external system that may receive this message, you configure the appropriate message XSL and routing information.

Because the outbound message type is associated with a business object, your implementation can easily create outbound message records from a script using the **Invoke business object** step type. Such a script would do the following:

- Determine the appropriate outbound message type and external system based on business rules
- Access the data needed to populate the message detail
- Populate the fields in the schema and use the **Invoke business object** script step type for the outbound message type's business object to store the outbound message.
- The resulting outbound message ID is returned to the script and the scriptwriter may choose to design a subsequent step to store that ID as an audit on the record that initiated this message.

The following topics provide more information about functionality supported by outbound messages.

- Polling Outbound Messages Using OSB
- Batch Message Processing
- Real Time Messages
- Designing the System for Outbound Messages
- Configuring the System for Outbound Messages
- Managing Outbound Messages

## Polling Outbound Messages Using OSB

If the outbound message that needs to be sent to an external system can be sent as an asynchronous message (in 'near real time'), the process initiating the outbound message


Copyright © 2026, Oracle and/or its affiliates.

should create a record in the outbound message staging table. Oracle Service Bus (OSB), is the recommended tool to use to process outbound messages in near real-time.

Outbound messages that should be processed by OSB should be configured with a processing method defined as **SOA** for the external system / outbound message type. No other information is required for defining outbound message types that are processed by OSB.

For the OSB part of the processing, the product provides a custom transport: OUAF Outbound Message that may be used by an implementation to define messages to process and how to process them. This transport processes outbound messages in order of the priority defined on the outbound message type.

This section provides an overview of steps required to develop OSB integrations for outbound messages created by your product.

Before developing OSB integrations, a developer should be familiar with OSB development such as creating proxy services, business services, and message flow/routing. These terms are defined as follows:

**Proxy Service:** In OSB, a Proxy Service is the entity that processes a given type of message and routes it to a Business Service. A separate proxy service should be defined for each type of outbound message. If a given outbound message type may be routed to different external systems, it is the responsibility of the proxy service to query the external system defined on the outbound message and invoke the appropriate business service (see below). If any transformation is required prior to routing a message to a business service, it is the proxy service's responsibility to perform the transformation.

**Business Service:** In OSB, a Business Service is an entity that receives a message from OSB and routes it to the appropriate destination. This should not be confused with the Business Service object provided in the product in the configuration tools.

**NOT_SUPPORTED**

Refer to the technical reference paper *OSB Integration* for more information.

## Batch Message Processing

Your implementation may be required to send messages to the same destination as a single XML file with multiple messages include. The following points describe this logic:

- The individual messages that should be grouped together must have a processing method of **batch** on the external system / outbound message type record. The appropriate batch code that is responsible for grouping the messages must also be provided.
- A separate "consolidated message" outbound message type should be configured for the external system with a processing method of **SOA**.
- When outbound message records are created for the individual messages, the batch code and current batch run number are stamped on the record.
- When the batch process runs it is responsible for building the XML file that is a collection of the individual messages. This batch process should include the following steps:
  - Format appropriate header information for the collection of messages
  - Apply the individual message XSL to each message before including the message


Copyright © 2026, Oracle and/or its affiliates.

  - Insert a new outbound message for the external system with the "consolidated message" outbound message type.
- The consolidated message is ready to be processed by Oracle Service Bus.

> **Note**
>
> **No process provided**. The system does not supply any sample batch job that does the above logic.

## Real Time Messages

The system supports the ability to make web service calls, i.e. sending real time messages, to an external system.

The system supports special functionality for sending an Email message real-time. Refer to Sending Email for more information.

For other types of real-time messages, the system also uses outbound message type and external system configuration to format and route the message. When defining the configuration for real time messages, an additional step is required to define the mechanism for routing the message using a message sender. The system supports routing messages via HTTP and via JMS. Note that for HTTP routing, the system also supports sending the message using a JSON format.

Just like near real-time messages, initiating a real-time outbound message may also be done from a script. When a real time message is added, the system immediately routes it to the external system. If the external system provided a response message back, the system captures the response on the outbound message. If the outbound message type for the external system is associated with a response XSL it is applied to transform the response. In this case the system captures the raw response as well on the outbound message. Note that the outbound message BO should be configured to capture a response XML in its schema.

Any error (that can be trapped) causes the outbound message to be in a state of **Error**. It is the responsibility of the calling process to check upon the state of the outbound message and take a programmatic action. When the outbound message state is changed back to **Pending** the message will be retried.

The base package provides two business services: Outbound Message Dispatcher (`F1-OutmsgDispatcher`) and Outbound Message Mediator (`F1-OutmsgMediator`) that further facilitate making web service calls. Both business services are similar, allowing the calling script to configure the following behavior (with differences noted):

- Whether or not exceptions encountered while sending the message are trapped. Trapping errors allows the calling script to interrogate any errors encountered and take some other programmatic action.
- Whether or not the sent message is persisted as an actual outbound message record.
  - If a persisted message is desired, the recommendation is to use the Outbound Message Dispatcher. This business service creates the message using standard BO processing, relying on the outbound message logic to route the message and store the record. The message is routed after the BO pre-processing algorithm and after the record is persisted but before the BO post-processing and audit plug-ins are executed. If you need to send the outbound message ID as part of the message, refer to Capturing the Outbound Message ID in the Message for more information.


Copyright © 2026, Oracle and/or its affiliates.

- If the message should not be persisted, then the recommendation is to use the Outbound Message Mediator. As mentioned, the Outbound Message Dispatcher creates the outbound message record, relying on the outbound message logic to route the message. If it should not be persisted, it is subsequently deleted. In contrast the Outbound Message Mediator executes the BO pre-processing algorithms and then routes the message explicitly without creating a message record. It is more efficient for scenarios that don't require persistence. Note that the Outbound Message Mediator also supports persistence, but it does so by creating the records without using BO processing. This is not recommended. The Dispatcher is the better option if persistence is desired.

Refer to the descriptions of the two business services for more information.

- OAuth Configuration

## OAuth Configuration

This section describes the various OAuth Grant Types supported by the system and highlights any special steps needed when using the grant type for a given use case. The information here is used to correctly configure the message sender context for senders that need this authorization type.

The following context entries apply to all OAuth grant types.

- **OAuth Access Token URL**
- Either the **OAuth Client Credentials Key Ring** or the **OAuth Client ID** and **OAuth Client Secret**. Refer to OAuth Configuration for more information about defining OAuth key rings.
- **OAuth REST API Scope**
- **OAuth Token Request Details in Body**

### Client Credentials

The client credentials grant type requests the access token using the client ID and secret credentials as defined above.

### JWT Assertion

The system supports both the JWT client assertion grant type and the JWT user assertion grant type. Client assertion proves the identity of the application making the request. User assertion, is for use cases where the specific application end-user is making the request.

Besides the context values applicable for all OAuth grant types, the JWT grant types also requires the following:

- **OAuth Signature Key Ring**. This defines the public / private signature key pair used to sign the assertion. Note that a prerequisite step required for contacting Oracle Identity Cloud Service is to upload a certificate based on this key.
- OAuth JWT Audience

# Designing the System for Outbound Messages

The following sections describe the setup required when using Outbound Messages to communicate with an external system. The configuration walks you through the steps to configure a single external system and all its messages.

- Define the Outbound Message Business Object and Type


Copyright © 2026, Oracle and/or its affiliates.

- Real-Time Message Configuration
- Define the External System and Configure the Messages
- Outbound Message Schema Validation

## Define the Outbound Message Business Object and Type

The product supplies many outbound message Business Objects along with Outbound Message Types for out of the box functionality.

In addition, implementations may need to define configuration for custom outbound messages. For each outbound message that must be sent to an external system, create a business object for the outbound message maintenance object. Using the business object's schema definition, your implementation defines the elements that make up the XML Source field (XML_SOURCE). These are the elements that are the basis for the XML message. XSL transformations may be applied to this XML source to produce the XML message.

If your integration is real-time and a response is expected, the Outbound Message business object should also map to the XML Response field (XML_RESPONSE).

- You may decide to capture the response as is and define the element as “raw”. For example

  ```
  <responseDetail mapXML="XML_RESPONSE" type="raw"/>
  ```

  In this scenario, a Response XSL may or may not be needed.
- Alternatively, if the details of the response are needed, you may define specific elements for the response. For this option, depending on how the integration is designed, a Response XSL may be needed to transform the response into the expected XML format.

Once you have your business object and schema, define an outbound message type for each unique outbound message.

### Referencing an XSL

The system supports defining the XSLs as a managed content record. For backward compatibility, the system supports defining the XSL as a file in the file system. This is a system wide setting, defined using a feature configuration option. The feature type is **External Messages** and the option type is **XSL Location**. Set the value to `F1FL` to support XSL in the file system. The file location is configured using a system property. To support the XSL in managed content, no option is needed as this is the default. You may explicitly define the value of `F1MC` if desired.

### Capturing the Outbound Message ID in the Message

If your integration would like to use the system generated Outbound Message ID as a unique identifier with the external system, the following configuration is needed:

- Define an element within the XML Source that should be populated with the system generated outbound message.
- Configure a BO Option on the outbound message’s business object using the Option Type **Outbound Message ID XPath** and set the Option Value to the XPath of the element defined to capture the ID.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This functionality is only applicable if the outbound message is persisted.

### Support for Dynamic URLs

The product supports the ability to build a dynamic URLs. These are cases where the URL requires information determined at runtime. This is supported with a combination of BO schema definition, URL configuration and appropriate code when creating the outbound message. The following points highlight the steps needed to support this functionality.

- When defining the Message Sender's URL, use the syntax `${pathParms}` in the location of the URL where runtime information must be inserted. For example: `http://[SERVER_NAME]:1000/rest/services/${pathParms}`
- Include the data area `F1-OM-DynamicConfig` (Outbound message dynamic configuration) in the schema of the business object for the outbound message. This data area includes elements for **pathParms** and **queryParms**.
- In the code that creates the outbound message, populate the **pathParms** and if applicable **queryParms** elements with the appropriate information. The system will build the URL by plugging in the **pathParms** element value followed by a question mark, followed by the **queryParms** element value into the `${pathParms}` location in the URL.

The following is an example of run time values.

```
<dynamicConfiguration>
 <pathParms>job/1234</pathParms>
 <queryParms>firstName=John&lastName=Doe</queryParms>
<dynamicConfiguration>
```

### Dynamic HTTP Headers

HTTP headers are typically static and the HTTP message senders support defining this static information as sender context. However, it is possible to support HTTP header information that may differ based on the individiual message and can only be provided at runtime. The following points highlight the steps needed to support this functionality.

- Include the data area `F1-OM-DynamicConfig` (Outbound message dynamic configuration) in the schema of the business object for the outbound message. This data area includes the list element **httpHeaders**, which supports a list of **httpHeader** elements.
- In the code that creates the outbound message, populate each 'header-key:header-value' pair that you want included in the HTTP header in the outgoing message. For example:

  ```
  move 'Lang:CHN' to "CM-RealTimeMessage/dynamicConfiguration/+httpHeaders/
  httpHeader";
  ```

### Add SOAP Header Parameters at Runtime

The product supports the ability to add SOAP header parameters to an external message at runtime. The following points highlight the steps needed to support this functionality.

- Include the data area `F1-OM-DynamicConfig` (Outbound message dynamic configuration) in the schema of the business object for the outbound message. This data area includes the element **soapHeaders** .


Copyright © 2026, Oracle and/or its affiliates.

- In the code that creates the outbound message, populate the **soapHeaders** element with the self contained XML to add to the SOAP header section of the outgoing SOAP request.

## Real-Time Message Configuration

Messages are routed to an external system real-time using the outbound message dispatcher or using the real-time send email business service. The system supports routing messages using HTTP and routing messages using JMS. In addition, there is a special type of message sender used for sending emails. The following sections highlights the supported real-time communication and the configuration needed for each.

### Email Messages

For sending emails, the following configuration is needed:

- Define a message sender configured for email. Senders of this type should be configured with the **RTEMAILSNDR** class. The sender context is used to configure the connection information for the connecting to the SMTP server.
- This sender may be defined as the default email sender on the message option table. Alternatively, the message sender can be provided to the business service as input. Refer to Sending Email for more information.

### Outbound Messages

For other outbound messages that are routed using the real-time outbound message dispatcher, a message sender must be configured to define how to route the message. The following points highlight more detail related to this configuration.

Determine the communication mechanism prior to configuring the sender.

- When routing the message using JMS, the following configuration must be defined
  - Define an appropriate JNDI Server that indicates where the JMS resources are located.
  - Define a JMS Connection to define additional configuration needed for the connection.
  - Define the JMS Queue or JMS Topic to define the queue or topic to use.
- When communicating using a JSON format, determine the method to use to convert the XML to JSON. The desired method is driven by how the request should be sent.
  - If choosing the **Base JSON Conversion** method, if XSL transformation needs to be applied prior to the conversion to JSON, then the target XML Request Schema must be defined (using a data area) so that the conversion logic knows the format of the XML it is converting. The XSL is applied to the outbound message's XML Source, resulting in the defined XML Request Schema, which is then converted to JSON. If XSL transformation is not required, then the outbound message's XML Source is converted to JSON.
  - If choosing the **Rootless JSON Conversion** method, the group element mapped to the XML source field is removed by the conversion, resulting in a rootless JSON request document.
  - If the XML source on the outbound message can be converted to JSON using an XSL, then the **XSL Transformation** method may be chosen.
  - You may also choose to convert the XML Source to JSON via the **Standard API Conversion** method (using a Jettison library). With this method, an XSL may optionally be provided. The conversion will be performed on the transformed XML.


Copyright © 2026, Oracle and/or its affiliates.

  - For the response, if the outbound message BO defines detailed elements for the XML Response field, then the JSON should be converted to this format.
    * If your conversion method is **Base JSON Conversion**, then if the JSON response cannot be converted directly to the XML Response elements on the outbound message BO, then define a Response Schema (data area) that represents the results for the base JSON conversion. In addition, define an XSL that can transform the response from the converted XML to the XML format expected on the BO.
    * If your conversion method is **Rootless JSON Conversion**, the response document is assumed to be rootless. The group element mapped to the XML response field is added by the conversion process, resulting in a well formed XML response document.
    * If the conversion method is **Standard API Conversion** or **XSL Transformation**, the standard API is used to convert JSON to XML. An XSL may be defined to convert the response to the XML Response if needed.
  - If the outbound message BO defines a "raw" element to capture the response, then a response schema and XSL are not necessary. In this case, the system will perform a JSON to XML conversion using the **Standard API Conversion** method (regardless of the conversion method defined) and the result is captured in the XML response.
- For HTTP senders including JSON senders, the system provides the following support for sending messages secured by OAuth authentication:
  - Using Oracle Web Services Manager (OWSM). The system provides a pre-configured set of policies for OAuth (`F1-0AUTH`) using a special extendable lookup. Note that the values of this policy set defines a specific CSF key repository that the implementation should use for capturing its CSF keys. In addition, there is a substitution value defined for the token URI: @F1_OAUTH2_URI@. Configure the appropriate URI for this implementation as described in URI Substitution. By default the system does not support additional policy sets to be defined. If your implementation requires a different policy set, contact support.
  - You may provide OAuth related settings as part of the message sender configuration for a REST API.

Define a message sender configured for each appropriate routing method. The invocation type should be configured as `Real-time`. For routing via HTTP, use the **RTHTTPSNDR** - HTTP sender class. For routing via HTTP with SOAP format automatically applied, use the **SOAPSNDR** - HTTP SOAP sender class. For routing via HTTP using JSON format, use the **RTJSONSNDR** - JSON sender class. For routing via JMS, use the **RTJMSQSNDR** - JMS queue sender class or **RTJMSTSNDR** - JMS topic sender class and configure the JMS Connection and JMS Queue or JMS Topic. Use the sender context to configure the required values for connecting to the appropriate destination.

> **Note**
>
> Refer to Support for Dynamic URLs for configuration needed to support dynamic URLs when sending an outbound message. There is specific configuration expected when defining the URL on the Sender in order to support dynamic URLs.

Configure the external system / outbound message type collection. The processing method defined for the external system / outbound message type must be `Real-time`.


Copyright © 2026, Oracle and/or its affiliates.

### Define the External System and Configure the Messages

Define an external system and configure the valid outgoing messages and their method of communication (processing method). Refer to Outbound Messages for more information.

### Outbound Message Schema Validation

The outbound messages that are generated by the system should be well formed and valid so that they do not cause any issues in the external system. To ensure this validity you may configure the system to validate messages before they are routed to their destination. Note that the validation is applied just before communication with the sender and therefore after any Request XSL has been applied.

- Define a directory where the valid W3C schemas are located using the Message option **Outbound Message Schema Location**.
- Each external system message must indicate the appropriate W3C schema to use for validation.

You may turn on or off this validation checking using the Message option **Schema Validation Flag**.

This feature is not supported for cloud implementations.

## Configuring the System for Outbound Messages

The following sections describe the setup required when using Outbound Messages to communicate with an external system.

- JNDI Server
- JMS Connection
- JMS Queue
- JMS Topic
- Message Sender
- Defining Outbound Message Types
- External Systems
- Message Option

### JNDI Server

If using JMS to communicate outbound messages, define a new JNDI Server. Open **Admin**, and then **Integration**, and then **JNDI Server**.

**Description of Page**

Enter a unique **JNDI Server** and **Description**.

Indicate the Provider URL to indicate the location of the JNDI server. A variable may be used in place of all or part of the URL. The variable must be predefined in the substitution variable property file. The value here should enclose the variable name with @. Refer to Referencing URIs for more information.

Indicate the **Initial Context Factory**, which is a Java class name used by the JNDI server provider to create JNDI context objects.


Copyright © 2026, Oracle and/or its affiliates.

## JMS Connection

To define a JMS Connection, open **Admin**, and then **Integration**, and then **JMS Connection**.

**Description of Page**

Enter a unique **JMS Connection** and **Description**.

Indicate the **JNDI Server** to be used. Refer to JNDI Server for more information.

Use the **JNDI Connection Factory** to indicate the lookup keyword in the JNDI server used to locate the JMS connection.

**Where Used**

## JMS Queue

To define your JMS Queue values, open **Admin**, and then **Integration**, and then **JMS Queue**.

**Description of Page**

Enter a unique **JMS Queue** and **Description**.

Enter the **Queue Name** as defined in the JNDI server. This is the JNDI lookup name identifying the queue.

Use the **Target Client Flag** to indicate whether or not the target client is **JMS** or **MQ**.

Select the **JNDI Server** where the queue is defined. Refer to JNDI Server for more information.

## JMS Topic

To define your JMS Topic values, open **Admin**, and then **Integration**, and then **JMS Topic**.

**Description of Page**

Enter a unique **JMS Topic** and **Description**.

Select the **JNDI Server** where the topic is defined. Refer to JNDI Server for more information.

Enter the **Topic Name** as defined in the JNDI server. This is the JNDI lookup name identifying the topic.

## Message Sender

The topics in this section describe the maintenance of a message sender

- Message Sender - Main
- Message Sender - Context

## Message Sender - Main

To define a new sender, open **Admin**, and then **Integration**, and then **Message Sender.**

**Description of Page**

Enter a unique **Message Sender** and **Description**.


Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! When adding new records, carefully consider the naming convention of the message sender code. Refer to System Data Naming Convention for more information.

Set **Invocation Type** to `Real-time`.

Indicate the **Message Class** for this sender, which indicates the method used to route the message. The valid sender classes are as follows:

| Message Class | Description |
|---|---|
| RTEMAILSNDR | Email sender. |
| RTHTTPSNDR | HTTP sender. |
| RTJMSQSNDR | JMS queue sender. |
| RTJMSTSNDR | JMS topic sender. |
| RTJSONSNDR | HTTP JSON sender. |
| SOAPSNDR | HTTP SOAP sender. |

> **ⓘ Note**
>
> For backward compatibility, the sender class HTTPSNDR is also supported, but should only be used if indicated by a specific integration configuration.

Indicate whether or not this sender is currently **Active**.

For JMS related senders, indicate whether the **Message Encoding** is **ANSI message encoding** or **UTF-8 message encoding**. (Note that for all other types of senders, the Character Encoding context type is used to configure encoding.)

If the Message Class is one that connects to a JMS Queue or JMS Topic, indicate the appropriate **JMS Connection**

> **NOT_SUPPORTED**
>
> Refer to JMS Connection for more information.

If the Message Class is one that connects to a JMS queue, indicate the name of the **JMS Queue** to define where the message is to be sent.

> **NOT_SUPPORTED**
>
> Refer to JMS Queue for more information.

If the Message Class is one that connects to a JMS topic, indicate the name of the **JMS Topic** to define where the message is to be sent.


Copyright © 2026, Oracle and/or its affiliates.

**NOT_SUPPORTED**

Refer to JMS Topic for more information.

## Message Sender - Context

The sender may require context information to define additional information needed by the system to successfully send outgoing messages. Open **Admin**, then **Integration**, then **Message Sender** and navigate to the **Context** page.

### Description of Page

Define the **Context Type** and **Context Value**, which contain parameters for senders when more information is required. See below for the supported context values for different types of senders.

### Email Sender Context

The email sender is used by the business service that sends email messages real-time.

An email sender must point to the Message Class **RTEMAILSNDR**. In addition, the following context records should be defined for senders of this type.

| Context Type | Description |
|---|---|
| SMTP Host name | The SMTP server host name. |
| SMTP Username | The user ID used to access the SMTP server. |
| SMTP Password | The password used to access the SMTP server. |
| SMTP From Address | The email address to use in the "from" field when sending the email. |
| SMTP TLS Enable | Set to 'true' or 'Y' to send emails using the Transport Layer Security (TLS) protocol. |
| Response Time Out | The amount of time the system should wait for a real time response. |

Once you have your email senders defined, configure on Message Option the default email sender to use when an explicit email sender is not defined.

### HTTP Sender

An HTTP sender is one that sends messages to an HTTP server using the HTTP protocol. HTTP senders should reference a Message Class of **RTHTTPSNDR**, **RTJSONSNDR** or **SOAPSNDR**. (Note that your implementation may use a specific integration that requires the **HTTPSNDR** message class. It should only be used if indicated that the integration's documentation.)

Various parameters are required to establish a session with the target HTTP server. You specify these parameters by defining a collection of context values for the sender. A set of context types related to HTTP variables is provided with the product. The following section describes the context types and where appropriate, indicates valid values.

Before defining the HTTP sender, you need to find out how the HTTP server on the other side expects to receive the request, and in particular, to answer the following questions:

- What is the address of the HTTP server?
- Is the HTTP server using a POST or GET HTTP method?


Copyright © 2026, Oracle and/or its affiliates.

- If the server is using POST, how are message contents passed? Does it use an HTTP FORM or does it pass the data in the body of an XML message?

| Context Type | Description | Values |
|---|---|---|
| HTTP URL1 - URL9 | Used to construct the URL of the target HTTP server.<br>Since the URL may be long and complex, you can break it into smaller parts, each defined by a separate context record. The full URL is built by concatenating the values in URL1 through URL9.<br>You may use substitution variables when entering values for URL parts. Note that the substitution string `@XMLMSG@` may be used for **GET** calls if an XSL has been applied to convert the message into HTTP GET parameters. It is useful if the HTTP Form is not applicable to the type of message.<br>Refer to Support for Dynamic URLs for configuration needed to support dynamic URLs when sending an outbound message. | |
| HTTP Method | The HTTP method used to send the message.<br>**Note**<br>The SOAP sender message class **SOAPSNDR** only supports the **POST** method. | **POST** or **GET** |


Copyright © 2026, Oracle and/or its affiliates.

| Context Type | Description | Values |
| --- | --- | --- |
| HTTP Transport Method | Specifies the type of the message. You can either send the message or send and wait for a response. For either method, be sure to set the HTTP Timeout parameter. | **Send** or **sendReceive** |
| HTTP Form Data | Used when the message is in the format of an HTML Form (`Content-Type: application/x-www-form-urlencoded`).<br>This context specifies the form parameters (data) that should be passed in the HTTP message. Since a form may have multiple parameters, you should add a context record for each form parameter.<br>The value of a form parameter takes the format of *x*=*y* where *x* is the form parameter name and *y* is its value.<br>If *y* contains the string @XMLMSG@ (case sensitive) then this string is replaced by the content of the service response XML message. The @XMLMSG@ string can be used in the HTTP Form Data or in the HTTP URL, but not in both.<br>If a context record of this type is defined for a sender, the sender uses the HTML Form message format to send the message even if @XMLMSG@ is not specified in one of the context records.<br>If a context record of this type is not defined for a sender, then the XML is sent with `Content-Type: text/plain`.<br>When using POST it is put in the HTTP message body.<br>Always required when using the GET method. If you are using the GET method and do not specify a Form Data context record, no message is transferred to the HTTP server.<br>You may use substitution variables when entering values for Form Data. | |


Copyright © 2026, Oracle and/or its affiliates.

| Context Type | Description | Values |
| --- | --- | --- |
| HTTP Login User | The HTTP server may require authentication. Add a context record of this type to specify the login user to use. | |
| HTTP Login Password | The HTTP server may require authentication. Add a context record of this type to specify the login password to use. | |
| HTTP Header | Sometimes the HTTP server on the other side may require the addition of HTTP headers to the message.<br>For each HTTP header that has to be specified you should add a context record with a value having the following format *x:y* where *x* is the header name and *y* is the value for the header<br>**Note**<br>The product also supports Dynamic HTTP Headers that may be included in the message at runtime. | |


Copyright © 2026, Oracle and/or its affiliates.

| Context Type | Description | Values |
|---|---|---|
| HTTP Time Out | Indicates the amount of time to wait for a connection to be established with the remote system. Be sure to indicate something here, otherwise users may wait a long time if an external system is not responding. | |
| Character Encoding | Indicates if the message should be encoded. The sender will add to the HTTP's content type header the string `;charset=x` where *x* is the value of this context and when sending the message it will encode the data in that encoding. | `UTF-8` or `UTF-16` |
| Response Time Out | The amount of time the system should wait for the remote system to send a response. | |

When sending messages, authorization is important. The following context types are related to authorization.

| Context Type | Description | Values |
|---|---|---|
| Sender Security Type | Indicate the desired security type to apply. (Note that additional security types are applicable only to the SOAP Sender. See below for more information.) | **BASIC** (HTTP Basic), **TEXT** (Username Token plain text), **DIGEST** - Username Token Digest, **OWSM** - OAuth Security via OWSM, **OAUTH** - OAuth Security. |
| OWSM Policy Set | Applicable only if the Sender Security Type is **OWSM**. Defines the policy set to apply. | Enter a valid value for the extendable lookup Set of Policies ( `F1-SetOfPolicies` ). The product provides the value `F1-OAUTH` that may be used here. |
| OAuth Grant Type | Applicable only if the Sender Security Type is **OAUTH** . Defines the OAuth specification method for acquiring an access token. Refer to OAuth Configuration for more information on defining OAuth security types. | `client_credentials`, `jwt_user_assertion`, `jwt_client_assertion` |
| OAuth Access Token URL | Used to construct the URL of the target authorization server. You may use substitution variables to construct the URL. | |
| OAuth Client ID | The generated identifier for the current application in the external system's authorization server. | |


Copyright © 2026, Oracle and/or its affiliates.

| Context Type | Description | Values |
|---|---|---|
| OAuth Client Secret | The secret paired with the OAuth Client ID as provided by the external system's authorization server. | |
| OAuth Client Credentials Key Ring | Defines the key ring used to store the OAuth Client ID and secret provided by the external system's authorization server. If a key ring is specified, the OAuth Client ID and OAuth Client Secret context entries are not allowed. | |
| OAuth JWT Audience | This is applicable when the OAuth Grant Type is one of the JWT assertion types. If populated, it is used as the JWT audience (aud) claim. If not provided, the OAuth Token URL is used as the audience. | Example: `https://identity.oraclecloud.com/` |
| OAuth REST API Scope | The authorization level requested from the token end point. | |
| OAuth Signature Key Ring | This key ring is applicable when the OAuth Grant Type is one of the JWT assertion types. It is used to sign the JWT request sent to the Identity Cloud Service. | |
| OAuth Token Request Details in Body | By default, the authorization token request includes the Client ID and Secret information in the message header. This context type may be used to indicate that this information should be sent in the body of the request instead. | Set this to 'Y' to send the Client ID and Secret information as part of the body of the authorization token request. |

### Real-time HTTP Sender

The following context type is only applicable to senders with the **RTHTTPSNDR** message class.

| Context Type | Description |
|---|---|
| Content Type | Populate a value here to override the `Content-Type` attribute in the HTTP header, which defaults to `text/xml` . |

### SOAP Sender

A SOAP sender is an HTTP sender that automatically adds support for the SOAP format. For this type of sender (message class of **SOAPSNDR**), besides the context values listed above, the following context entries may also be supplied.


Copyright © 2026, Oracle and/or its affiliates.

| Context Type | Description | Values |
|---|---|---|
| Message Namespace URI | Used to indicate a specific namespace to be included in messages for this sender. Note that this value is used only when the External Message link to this sender is configured with a **Namespace Option** of **Configured on Sender**. | |
| SOAP Insert Timestamp | Indicate whether a timestamp should be added. Default value is 'N'. | Y<br>or<br>N |
| SOAP Expiration Delay (in seconds) | Indicate an expiration delay to add to the timestamp. The default value is 60. | |
| Sender Security Type | Additional security types are supported for SOAP senders for specific third party integrations. | **CIM** - Uses an appropriate namespace and adds security information in the SOAP header specific to Landis+Gyr, MS_V30<br>- Uses an appropriate namespace and adds security information in the SOAP header specific to MultiSpeak v3.0, MS_V41R<br>- Uses an appropriate namespace and adds security information in the SOAP header specific to MultiSpeak v4.1.**ITRON** - Uses appropriate SOAP header information security information in the SOAP header specific to Itron. |

> **Note**
>
> Refer to Add SOAP Header Parameters at Runtime for information about dynamically including SOAP Header parameters when sending a message.

### JMS Sender

A JMS sender is one that sends messages to a JMS queue or JMS topic. JMS senders should reference a Message Class of **RTJMSQSNDR** or **RTJMSTSNDR**, respectively.

The following parameters are used to connect to the JMS resource.

| Context Type | Description | Values |
|---|---|---|
| JMS Message Type (Bytes(Y)/ Text(N) | Indicates whether the data is sent as a bytes message or as a text message. | Y<br>or<br>N |


Copyright © 2026, Oracle and/or its affiliates.

| Context Type | Description | Values |
|---|---|---|
| JMS User Name | Enter the user name to connect to the JMS resource. | |
| JMS User Password | Enter the password to use to connect to the JMS resource. | |
| JMS Header | If JMS header values are required for the message, use this context type.<br>For each JMS header that has to be specified, add a context record with a value having the following format *x:y* where *x* is the header name and *y* is the value. | |

### Application Variables

Some APIs may require application identifying variables to be included in the message payload. These values are typically available at provisioning time and need to be captured in relation to these API configurations.

> **Note**
>
> This type of context is not involved in the process of orchestrating and routing the API call. It needs to be incorporated into the payload by the application logic that composes the message using the "Get Application Variables (F1MsgVars)" service script.

The following parameters may be used to capture application variables.

| Context Type | Description | Values |
|---|---|---|
| Application ID | Identifies the current environment for this API. | Determined at provisioning time by the API provider. |
| Tenant ID | Identifies the current tenant for this API. | Determined at provisioning time by the API provider. |

## Defining Outbound Message Types

Refer to Outbound Messages for an overview of this functionality.

Use this page to define basic information about an outbound message type. Open this page using **Admin Menu**, then **Integration**, then **Outbound Message Type.** You are brought to a query portal with options for searching for outbound message types.

> **Note**
>
> This page is not available if the **External Message** module is turned off.

Once an outbound message type has been selected, you are brought to the maintenance portal to view and maintain the selected record.


Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page.

- **Outbound Message Type.** This zone provides basic information about an outbound message type. Besides the code and description, the outbound message type captures the outbound message business object and the relative priority of the outbound message. The priority is only applicable if the outbound message is processed using OSB. In addition, if the outbound message type is linked to any web service categories, they are displayed.

> ⚠ **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the outbound message type code. Refer to System Data Naming Convention for more information.

- **External System Links.** This zone is visible if the outbound message type is linked to any external systems.

Note that in addition to standard actions available on this portal, there is also a special **Add to Category** button in the page action area. Click this button to link an outbound message type to one or more web service categories.

The following zones may appear as part of the portal's **Help Text** tab page.

- **Help Text Fields.** This zone maintains help text information for fields associated with the outbound message business object.

> ⓘ **Note**
>
> Help text is not translated.

## External Systems

Use this page to define an external system and define the configuration for communication between your system and the external system.

- External System - Main
- External System - Template Use

## External System - Main

Open this page using **Admin**, and then **Integration**, and then **External System.**

> ⓘ **Note**
>
> This page is not available if both the **External Message** and the **Open Market Interchange** modules are turned off.

### Description of Page

Enter a unique **External System** and **Description**.


Copyright © 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Important! When adding new records, carefully consider the naming convention of the external system code. Refer to System Data Naming Convention for more information.

Use the field **Our Name In Their System** to specify the identity of your organization (i.e., your external system identifier) in the external system.

> **ⓘ Note**
>
> The workflow process profile and notification download profile are only applicable to products that support workflow and notification. They and are not visible in the product if the **Open Market Interchange** module is turned off.

If this external system sends inbound communications through notification upload staging, the type of workflow process that is created is controlled by the sender's **W/F (Workflow) Process Profile**.

If you send notifications to this external system, select a **Notification DL** (download) **Profile** that is used to define the configuration of the outgoing messages.

> **ⓘ Note**
>
> The remaining fields are not visible if the **External Message** module is turned off.

Set **Usage** to **Template External System** for external systems whose outbound message type configuration is inherited by other external systems.

If the outbound message type configuration should be inherited from a template external system, define the **Template External System**. If this field is specified, the outbound message type collection displays the data defined for the template system as display-only.

The **Outbound Message Type** accordion contains an entry for every type of outbound message defined for this external system. For each type of outbound message identify its **Outbound Message Type**.

Define the **Processing Method** for messages of this type. Valid values are **Batch**, `Real-time`, **SOA** and **XAI**.

Define an appropriate **Message Sender** if the processing method is **XAI** or `Real-time`.

Refer to Exchanging Date / Time Information With External Systems for more information about date/time formats.

**Namespace Option** is used when your message should include a namespace in the resulting XML. The valid option are **Standard Namespace** and **Configured on Sender**. If the value is Standard Namespace, the system will generate a namespace for the resulting WSDL with the following value: `http://ouaf.oracle.com/outbound/boName`, where boName is the business object code of the outbound message (configured on the outbound message type defined for this profile). If the value is Configured on Sender, then the value of the namespace is taken from the sender context with context type **Message Namespace URI**.

Define an appropriate **Batch Control** if the processing method is **Batch**.


Copyright © 2026, Oracle and/or its affiliates.

If the message sender is one with a message class of **RTJSONSNDR**, indicate the **JSON Conversion Method**. The valid values are **Base JSON Conversion**, **Standard API Conversion** and **XSL Transformation**. Refer to Real Time Message Configuration for more information about these methods and the additional configuration that may be applicable.

If the **JSON Conversion Method** is **Base JSON Conversion**, the **Request Schema** is enabled. Populate a data area that defines the schema for the XML format to convert the outbound message's BO schema to prior to performing the JSON conversion. Refer to Real Time Message Configuration for more information.

The **Message XSL** is the schema used to transform information from the format produced by the system to a format understood by the sender, who receives a message of this type. This is not applicable for Processing Method of **SOA**. Refer to Referencing an XSL for information about configuring where the XSL is defined.

Enter the file name of the appropriate **W3C Schema** if you want to validate the message built for outbound messages for this external system / outbound message type prior to being routed to their destination. Refer to Outbound Message Schema Validation for more information. This is not applicable for Processing Method of **SOA**. This feature is not available for cloud implementations.

If the **JSON Conversion Method** is **Base JSON Conversion**, the **Response Schema** is enabled. Populate a data area that defines the schema for the XML format that the JSON message is initially converted to. The XML is then converted to BO XML. Refer to Real Time Message Configuration for more information.

**Response XSL** will only be displayed when the processing method is `Real-time`. Refer to Real Time Messages for more information on how it is used. Refer to Referencing an XSL for information about configuring where the XSL is defined.

## External System - Template Use

If you are viewing an external system whose usage is a **Template External System**, use this page to view the other external systems that reference this one. Open this page using **Admin**, and then **Integration**, and then **External System** and then navigate to the **Template Use** tab.

**Description of Page**

The tree shows every external system that references this external system as its template.

# Message Option

The Message Option page defines various system settings used by the system when processing external messages.

To define options for your environment, open **Admin**, and then **Integration**, and then **Message Option**.

**Description of Page**

The following options are supported.

| Option | Description | Option Name |
|---|---|---|
| Default Email Sender | This is the default Message Sender used for sending e-mails when no explicit Message Sender is specified. | `defaultEmailSender` |


Copyright © 2026, Oracle and/or its affiliates.

| Option | Description | Option Name |
|---|---|---|
| Default User | The default user is used by the system to access your product when no other user is explicitly specified. | `defaultUser` |
| Email Attachment File Location | This is the default location of e-mail attachment files. If not specified, the e-mail service provided with the product assumes a full path is provided with each attachment file. | `emailAttachmentFileLocation` |
| Email XSL File Location | This is the default location of e-mail XSL files. If not specified, the e-mail service provided with the product assumes a full path is provided to an XSL file as part of an e-mail request. | `emailXSLFileLocation` |
| Messages Language | The default language to use for the messages. | `language` |
| Outbound Message Schema Location | Enter the full path of the virtual directory where valid W3C schemas are stored if your implementation wants to validate outbound message schemas. For example: http:/localhost/cisxai/schemas. | `xaiOuboundSchemaLoc` |
| Schema Validation Flag | Enter `Y` to turn on schema validation for outbound messages. Enter `N` to turn this off. | `xaiSchemaValidationCheck` |

Note that the following options are no longer applicable.

| Option |
|---|
| To Do Type for Inbound JMS Message Errors |
| To Do Type for Outbound Message Errors |
| XSD Compliance |

# Managing Outbound Messages

Use this page to view information about outbound messages.

- Outbound Message - Main
- Outbound Message - Message
- Outbound Message - Response

## Outbound Message - Main

Open this page using **Menu**, and then **Integration**, and then **Outbound Message**.


Copyright © 2026, Oracle and/or its affiliates.

**Description of Page**

**Outbound Message ID** is the system-assigned unique identifier of the outbound message. These values only appear after the outbound message is added to the database.

The **Processing Method** indicates whether this record will be processed by a **Batch** extract process or `Real-time`. (The value of **XAI** is no longer supported). The value defined on the external system / outbound message type collection populates this value.

When records are created with a processing method of **Batch**, the system sets Extract to **Can Be Extracted**. Change the value to **Not to be extracted** if there is some reason that processing should be held for this record.

For records with a processing method of **Batch**, **Batch Control** indicates the process that will extract this record. This value is populated based on the on the external system / outbound message type's value. **Batch Number** indicates in which batch run this record was extracted or will be extracted.

The **Retry Count** is no longer applicable..

The **Creation Date** indicates the date that this record was created.

The status is no longer applicable.

## Outbound Message - Message

Use this page to view the XML source used to build an outbound message. Open this page using **Menu**, and then **Integration**, and then **Outbound Message** and then navigate to the **Message** tab.

**Description of Page**

The **XML Source** is displayed.

If a message XSL is defined on the external system / outbound message type record linked to this outbound message, the **Show XML** button is enabled. Click this button to view the XML that is a result of applying the Message XSL to the XML source.

## Outbound Message - Response

Use this page to display the XML response. Open this page using **Menu**, and then **Integration**, and then **Outbound Message** and then navigate to the **Response** tab.

**Description of Page**

The **XML Response** and optionally the **XML Raw Response** is displayed.

XML Response displays the response data from the system called by the real-time message. If a response XSL is defined on the external system / outbound message type record linked to this outbound message, a transform is performed and the XML Raw Response displays the original, unchanged response.

# Web Service Adapters

The base product provides a configuration object called Web Service Adapter that is used to help build configuration objects to allow for functionality in the system to initiate a web service call from within the system. A Web Service Adapter provides the following functionality:

- WSDL (web service description language) import. An implementer can use the WSDL import functionality to read the details of a WSDL into the system


Copyright © 2026, Oracle and/or its affiliates.

- Internal API generation. The system generates internal data areas that have two main purposes: they provide the API for custom code to define the appropriate input and they provide output data for the web service call using Oracle Utilities Application Framework schema language. In addition, the web service dispatcher uses element mapping defined in the data areas to transform the internal XML into the structure expected by the external system as described in the WSDL.
- Defines the URL needed to perform the web service call at runtime.
- Understanding Web Service Adapters
- Setting Up Web Service Adapters

# Understanding Web Service Adapters

The following topics describe the system functionality in more detail.

### Importing a WSDL

Configuring a Web Service Adapter starts by identifying the WSDL (the web service description language document used to define the interface) that will be provided by the external system. The following steps describe the base product functionality provided to allow a user to import a WSDL.

- Navigate to the **Web Service Adapter** page in add mode and select the appropriate base business object.
- Enter a meaningful Web Service Name and appropriate descriptions.
- Provide the URL of the given WSDL.
- Click **Import** to retrieve the details of the WSDL. The system then parses the WSDL details and populates the WSDL Service Name, WSDL Source, WSDL Port, URL and a list of Operations (methods) defined in the WSDL.
- Determine which Operations should be **active** based on the business requirements for invoking this web service. **Active** operations are those that the implementation is planning to invoke from the system. These require appropriate request and response data areas generated for them. The following section provides more information about that.
- Specify the appropriate Security Type to configure the type of security to use when invoking this web service.
- Click Save.

At this point, a web service adapter record is created in pending status. The next step is to generate the request and response data areas for the operations configured as active.

### Generating Request and Response Data Areas

Each **active** operation for the web service adapter requires a pair of data areas, request and response, that represent the request and response XML messages for the operation.

The base product provides steps to generate the data areas as follows:

- As described in the Importing a WSDL section above, the operations listed in the WSDL are generated for the web service adapter and the implementer should indicate which operation to activate.
- After saving the **pending** web service adapter, the display lists all the active operations and for each one includes a **Generate** button.


Copyright © 2026, Oracle and/or its affiliates.

- After clicking **Generate** for an operation, a window appears where the names of the new Request and Response Data Areas may be defined. Click **Save** to generate the data areas.

The generated data areas provide the API for the implementer to use when implementing the web service call in an appropriate algorithm or service in the system. The data areas contain the appropriate mapping from the elements that the implementer works within the code that invokes the web services and the WSDL definitions.

To facilitate generating the request and response data areas, the base product invokes a special business service used to create the appropriate mapping. The business service is defined as a BO option on the Web Service Adapter business object. This allows an implementation to provide a custom business service to further enhance the request and response mapping where appropriate.

> **Note**
>
> **Generated data areas.** It is possible to edit and modify the generated data areas after they are created. An implementer can change element names or remove unneeded elements if desired. Manually changing the generated data areas must be done only when absolutely necessary. This is because the system is not able to validate manual changes and issues with the data areas would only be detected at run time.

### Activating Web Service Adapters

The business objects provided by the base package for web service adapters include a simple lifecycle of **Pending** and **Active**. Configure the web service adapter and its data areas while in **Pending** status and activate it when it is ready to be implemented in the appropriate system functionality.

### Invoking Web Services

To make a call to a web service using a web service adapter, the system has provided a Web Service Dispatcher business service (**F1-InvokeWebService**) to submit a web service call. The calling program is responsible for retrieving all the information to correctly populate the request data required by the web service call before invoking the business service.

> **Note**
>
> Refer to the detailed description of the business service for more information.

### Limitations

The following points highlight limitations associated with the types of web services that the system supports:

- It is possible for one WSDL document to contain definitions for several web services. The system currently supports only one port or service per WSDL document.
- It is possible for a WSDL to support multiple message patterns. The system currently supports only request / response.

## Setting Up Web Service Adapters


Copyright © 2026, Oracle and/or its affiliates.

Use the Web Service Adapter portal to define the configuration needed to communicate with an external system using a web service call. Open this page using **Admin**, and then **Integration**, and then **Web Service Adapter.** You are brought to a query portal with options for searching for web service adapters.

Once a web service adapter has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Web Service Adapter** zone provides basic information about the web service annotation type.

Please see the zone's help text for information about this zone's fields.

> **NOT_SUPPORTED**
>
> Refer to Understanding Web Service Adapters for information about common web service adapter functionality.

# Sending Email

The framework provides the ability to initiate an email from within the system. The following topics highlight the functionality available.

- Sending email "real time" using a specific business service. The framework provides a business service `F1-EmailService` that supports sending an email. The schema supports elements for all the information required to create an email real time. The SMTP information (host, user name, password and 'from' address) may be provided or may be defined on a message sender, that may be provided as input. In addition, the business service supports using a default message sender defined as a message option. Review the business service schema for information about the input elements.

  > **Note**
  >
  > **Retry setting**. An option in the system properties file allows your implementation to configure the number of times to retry (if any) if the SMTP server is unavailable. Refer to the server administration guide for more information.

- The business service supports sending attachments. The may either reference attachments from the file system (file name and content ID) or reference attachment records captured in the system's attachment table. Note that attachments in the file system are not supported in a cloud implementation.
  - For references to the attachment records, refer to Emailing Attachments for information about the location and size of attchment content and how that affects how attachments are included in the email.
  - For referencing attachments from the native file system, an entry must be defined for the file path in the File Storage Configuration extendable lookup. Refer to URI Substitution for more information on defining this configuration. The extendable lookup records are acting as a type of allowlist.
    The system supports defining a 'parent' directory in the file storage extendable lookup. When invoking F1-EmailService, providing a directory within the registered 'parent' directory is supported. For example, if you register the path D:\systemFiles\, at runtime, when invoking F1-EmailService, you can provide a file in the path D:\systemFiles\CMA.


Copyright © 2026, Oracle and/or its affiliates.

You may also use the `file-storage://XXXX` syntax described URI Substitution when providing the file location to the F1-EmailService.

> **Note**
>
> Validating attachments. If a Validate Email Attachment algorithm is plugged into the installation record, it is called to validate the attachments supplied, if applicable.

- Using an outbound message to send an email. This option allows for different variations as described in Outbound Messages.
  - Some emails may be created en masse (for example a large group of emails routed to users for a given set of To Do entries). In this case, the records can be created in the staging table for processing using OSB.
  - Messages may still be sent real time using one of two business services described in Real Time Messages. This option is an alternative to the dedicated email service described above when aspects of the outbound message functionality are needed, such as the ability to instantiate a record as an audit or to include additional logic via BO plug-ins as part of sending the email.

# Web Service Category

The product provides the ability to categorize web services that are defined in the system. The term web services refers to Outbound Message Types, Inbound Web Services, and for those implementations where they are applicable, XAI Inbound Services.

A given web service may be linked to more than one category. The product supplies web service categories and most base delivered web services are configured with appropriate categories. Implementations may define new categories and may link custom web services to base delivered categories or to custom categories. In addition, implementations may link additional categories to base delivered web services. However, implementations may not remove base delivered web services from base delivered categories.

If your implementation uses the Integration Cloud Service Catalog, the integration provides the categories for each web service that is sent to the catalog.

The web service category portal supports adding and removing web services from the category. In addition, the Outbound Message Type portal and Inbound Web Service portal each provide a page action button to **Add to Category**.

- Defining Web Service Categories

## Defining Web Service Categories

Refer to Web Service Category for an overview of this functionality.

Use this page to define basic information about a web service category. Open this page using **Admin**, and then **Integration**, and then **Web Service Category.** You are brought to a query portal with options for searching for web service categories.

> **Note**
>
> This page is not available if the **External Message** module is turned off.


Copyright © 2026, Oracle and/or its affiliates.

Once a web service category has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Web Service Category** zone provides basic information about a web service category.

> ⚠ **Caution**
>
> Important! When adding new records, carefully consider the naming convention of the web service category code. Refer to System Data Naming Convention for more information.

Once a web service category exists, an **Add Web Services** zone is provided to find and select web services to link to the web service category.

When viewing a category that already has web services linked to it, they are visible in the zone **Included Web Services** . This zone allows a user to remove a customer owned web service from the list.

# JMS Message Browser

The JMS Message Browser portal allows you to select a JMS queue and view messages currently in the queue.

In order for a JMS queue to be available on the portal, a message sender must be defined that is configured for the appropriate JMS queue with the credentials to connect to the queue.

- If your organization sends real-time outgoing messages to a JMS queue, this configuration would exist as per the details in Real-Time Message Configuration.
- If inbound web service messages are routed to the system via a JMS queue, no configuration is needed in the system. However, if you would like to view the messages in the queue in the JMS message browser portal, configuration for the JMS queues as described for outgoing messages is required.

Navigate to the portal using **Main**, and then **Integration**, and then **JMS Message Browser.**

The JMS Senders zone provides a list of configured message senders eligible for selection.

The JMS Message List zone is visible for the JMS Sender broadcast from the first zone. This zone supports selecting one or more records to delete from the queue. Use the message selector to limit the results to messages that satisfy the message selector. This uses standard JMS API message selector functionality. Refer to the zone embedded help for information about the supported syntax.

The JMS Message Details zone displays a message broadcast from the list zone.

# Oracle Integration Cloud Catalog

Oracle Integration Cloud (OIC) is an offering that serves as integration infrastructure for Oracle cloud solutions. The product provides an adapter for OIC to streamline integration between your edge application and OIC.

The product provides a mechanism (referred to as an adapter) for OIC to retrieve the REST catalog and another to retrieve the SOAP catalog from a given product. The expectation is that a given implementation would use either REST or SOAP and not both, but for those implementations that do support both classes of web services, OIC can retrieve both catalogs.


Copyright © 2026, Oracle and/or its affiliates.

For the SOAP catalog, the adapter retrieves the name, the source system, the WSDL location and namespace for each web service.

For the REST catalog, the adapter retrieves the name, the source system and the OpenAPI Specification for each web service.

It is possible that not every web service supported by an edge product is managed by OIC. In order to only include the appropriate web services in the adapter, configuration is needed to identify which web services to include.

The web service catalog is used to identify the records that should be retrieved by the adapter. Separate catalogs exist for REST and for SOAP

- For SOAP inbound messages, the system supports both the use of inbound web services and XAI inbound services that are deployed via IWS. Each IWS or XAI inbound service that should be included in the catalog must be flagged in the SOAP catalog. Note that only deployed services are returned to the catalogue.
- For REST inbound messages, each IWS that should be included in the catalog must be flagged in the REST catalog.
- For outbound message, the system requires the creation of an External System that includes each outbound message type that the external system receives. For outbound messages that are integrated through OIC, the external system itself will represent OIC. Rather than identifying each outbound message type to include in the catalog, only the external system needs to be flagged. The adapter will return web service information for all the outbound message types configured for the external system. If an implementation supports both REST and SOAP, separate external systems should be created to separately group the REST and SOAP outbound messages.

> **Note**
>
> The subordinate catalog functionality is no longer applicable and will be removed in a future release.

> **NOT_SUPPORTED**
>
> Refer to Maintaining the Web Service Catalog for more information.

- Web Service Catalog Configuration

# Web Service Catalog Configuration

The topics in this section describe the configuration needed in your edge application to integrate with OIC.

- Web Service Catalog Master Configuration
- Maintaining the Web Service Catalog

## Web Service Catalog Master Configuration

The **Service Catalog Configuration** (master configuration) record defines several system wide settings related to integrating with the service catalog.


Copyright © 2026, Oracle and/or its affiliates.

It includes configuration for the WSDL base URL for the SOAP catalog. Note that this element supports the functionality described in Referencing URIs. Note that for the REST catalog, the system will use the URL defined in the substitution variable `F1_OPEN_API_BASE_URL`

The subordinate server section is no longer applicable and will be removed in a future release.

For more information about specific fields in the master configuration, refer to the embedded help.

### Maintaining the Web Service Catalog

Refer to Oracle Integration Cloud Catalog for an overview of web service catalog functionality.

To add or remove services that are reported to the OIC catalog, navigate to the portal using **Admin**, and then **Integration**, and then **Web Service Catalog.**

The Catalog by Class zone provides the list of classes (REST and SOAP). Broadcast the web service class that you want to work with.

The Catalog zone provides a list of services that are currently in the catalog based on the web service class chosen. Users may use this zone to remove services from the catalog.

The Candidate Services zone provides a list of external systems and inbound web services that are not currently linked to the catalog. Users may use this zone to add services to the catalog. The inbound web services will be limited to those with the same web service class as the one broadcast. All external systems are shown regardless of the web service class because there is no web service class configuration on external systems.. The expectation is that implementers will create the external system to include either only REST or only SOAP services and will know which external system to include in which catalog. For the SOAP web service class, if XAI inbound services are applicable, they are also visible.

> **Note**
>
> For inbound SOAP web services and XAI inbound services, they may be selected for the SOAP catalog at any time. However, only deployed services are returned to the catalog by the OIC adapter.

## XAI Documentation Note

The XAI functionality is legacy functionality and not recommended for new implementations. The help topics are no longer provided.

# Integrations

The primary integration methods supported by products built on the Oracle Utilities Application Framework are (a) inbound and outbound files, and (b) inbound and outbound web services.

**File Based Integrations**

Files are uploaded to Object Storage or your local file system and processed via scheduled batch processes. Your specific product provides many out of the box upload processes. In addition, the system provides a plug-in driven upload process if you have specific upload requirements that are not met by base product processes. Refer to Uploading Records for more information.


Copyright © 2026, Oracle and/or its affiliates.

File-based extracts can be generated and made available for download and further processing. Several out of the box extract processes are provided by your specific product. Implementation-specific file processing and generation logic can be introduced using the plug-in driven Extract Record Processing.

**Web Services Integration** is supported through Inbound Web Services and Outbound Messages. Refer to those respective chapters for more detail.

This remaining topics in this chapter provide high level information about product integrations supported for all products that use Oracle Utilities Application Framework.

- Creating Application Links in External Applications
- LDAP Integration
- Oracle Identity Manager Integration
- Provision Users with SCIM Open Standard
- User Redirection
- Data Synchronization
- Operational Analytics
- Analytics Integration
- Business Flags

# Creating Application Links in External Applications

When integrating Oracle Utilities applications with other external applications, you can create links from external applications to specific portals and pages within the Oracle Utilities applications (also known as "deep linking"). These links typically leverage Business Process Assistant (BPA) scripts to help navigate the user to the appropriate target portal or page and to make provided parameter values available for use on the portal or page.

> **Note**
>
> This technique may also be used within the product to launch a new browser tab from within the application when it's important to maintain the current screen. For example, launching the GIS Viewer in Oracle Utilities Work and Asset Management by creating a new browser tab.

To do this, parameters are added to the application URL by appending a question mark (?) to the end and using the syntax "key=value". If you require more than one parameter, use an ampersand (&) to separate each key=value pair. There are several predefined parameter key values recognized by the application. In addition, specific parameters understood by the script may be provided. The following list highlights the predefined parameters.

- `location`=<navigation key> - If the exact page to launch is known, the navigation key value of that page can be part of the URL using 'location='. For example, if an approval process has sent an email to a user to approve an adjustment, the link configured in the email may include the adjustment's navigation key.
- `locationNavOpt`=<navigation option> - If the exact page to launch is known, the navigation option value of that page can be part of the URL using 'locationNavOpt='. For example, if an approval process has sent an email to a user to approve an adjustment, the link configured in the email may include the adjustment's navigation option. Assumption is


Copyright © 2026, Oracle and/or its affiliates.

that the navigation option provided is for a portal unless the URL explicitly defines "portal=false".

- `script=`<BPA script name> - In almost every use case, a script is needed to help navigate to the correct page and display the correct information. Even if a location is known and provided in the URL, if there is context data provided as a parameter, it is the responsibility of a script to move that parameter data from the URL into the page data model, so that the page can access the information and display the correct record. In cases where the target page is dynamic based on information provided by parameters, the script may be used to determine what page to launch.
- `portal=true` - When using the location parameter where the target page is a portal page and passing context data, use this parameter. The allows the system to know that the parameter data must be pushed into portal context.
- `initNav=false` - By default, if a script is provided but the location attribute is not provided, the system navigates to the user's home page prior to executing the script. If the script itself includes a step to navigate to a target page as one of its initial steps, the navigation to the home page is unnecessary and may degrade performance. This optional attribute may be included in the URL to bypass the home page.

> **Note**
>
> The system still requires a page to be launched for technical reasons. A blank portal with no zones is used for this purpose. Users may see this portal (called **Launching Application**) briefly before the navigation initiated by the script. In addition, this is the portal that the user will remain on if there are any errors in the script or if the script does not navigate anywhere.

- `minSidebar=true` - This option minimizes the sidebar area as part of launching the application. (The user will be able to maximize the sidebar themselves if needed). You would use this option to potentially improve the speed at which the application is launched. Whether this is useful will depend on the use case. If the reason for launching the application is to bring the user to a specific record to view details about that record, the sidebar may be providing valuable information about the record. In this case, it may be correct to display the sidebar by default (and not include this parameter in the URL). On the other hand if the use case is that a user is already in the application and clicks a link that opens a new browser tab to show the user specific detail about a chosen record, while keeping the original page open on the first tab, the sidebar may contain the same information already visible on the original tab. In this case, minimizing the sidebar may be beneficial.

> **Note**
>
> For backward compatibility, the system still supports the original parameter `minDashboard=true`, which has analogous functionality.

For example, the following URL illustrates launching the script `CM-StopAutoPay` for a given account, assuming the application URL for launching the system is http://system-server:1234/cis.jsp:

- **http://system-server:1234/cis.jsp?script=CM-StopAutoPay&ACCT_ID=1234512345&location=accountMaint**


Copyright © 2026, Oracle and/or its affiliates.

It doesn't matter in which order the parameters are provided. The system processes them in the correct order. For example, the following examples are processed by the system in the same way:

- **http://system-server:1234/cis.jsp?ACCT_ID=1234512345&script=CM-StopAutoPay&location=accountMaint**
- **http://system-server:1234/cis.jsp?ACCT_ID=1234512345&location=accountMaint&script=CM-StopAutoPay**

These parameters are kept in a common area accessible by any script for the duration of the session. To use these parameters on a script you may reference the corresponding `%PARM-<name>` global variables. In this example, after the system is launched any script may have access to the above account ID parameter value by means of the `%PARM-ACCT_ID` global variable. Also note, these parameters are also loaded into temporary storage (to continue the example, there would also be a temporary storage variable called `ACCT_ID` that holds the passed value).

### Determining the Application URL

When building the link from within application logic, the service script `F1-EnviURL` may be used the retrieve and build the application URL information.

### Navigate to a Given Record's Maintenance Portal

The product provides a script `F1-GotoPrtl` (Navigate to portal for an MO and key values) that may be used for any use case that simply navigates to the maintenance page for a given record and displays that record. This script is only applicable to records that are governed by a business object (and define a navigation option as a BO option). It requires the parameters `mo=` with the maintenance object code of the record and the parameters `pkValue1=` through `pkValue5=` for providing the key values. It looks up the appropriate portal navigation option from the record's BO, populates keys into the 'page data model' and navigates to the portal.

It is also recommended to include the `initNav=false` attribute when building a URL with this script. The example below navigates to the appropriate Migration Data Set portal for the given ID's business object.

- **http://system-server:1234/cis.jsp?script=F1-GotoPrtl&pkValue1=1234512345&mo=F1-MIGRDS&initNav=false**

# LDAP Integration

Organizations commonly use a Lightweight Directory Access Protocol (LDAP) security repository as a source of security credentials. The system provides support for importing users and groups from an external LDAP repository into the product to populate Users and User Groups in the system. Once imported, all user and group functions are available. You can resynchronize your LDAP users and groups at any time.

> **Note**
>
> **Import only**. The system currently supports importing LDAP information. Exporting users and groups from the system to LDAP is not provided.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional configuration. When importing new users and / or groups, additional configuration is needed in the base product. For example, after importing a new user group and its users, the user group configuration should be updated to define the valid application services for the user group. After importing a new user, additional configuration may be needed on the user such as valid To Do Roles, valid Home Page, etc.

> **NOT_SUPPORTED**
>
> Refer to Users and User Groups for more information about the application security and what it controls.

This section the functionality provided in the framework application that supports LDAP. Refer to the *LDAP Integration* technical reference paper for more information about typical steps related to the full integration.

- LDAP Integration Overview
- Configuring LDAP Integration

# LDAP Integration Overview

This topic provides a high level overview of the integration process.

At a high level, the base product provides a process to import user group and / or user definitions from and LDAP repository. This is a one way integration.

- When importing a user, if it is not already found in the system, it will be added; otherwise its attributes will be updated according to the imported information.
- When importing a user group, if it is not already found in the system, it will be added; otherwise its attributes will be updated according to the imported information.
- When importing a user, its user group links will be updated as per the information in the import file. In addition, if there are any user groups linked to the user that are not found in the system, they will be added (however, the other users linked to that group in the LDAP repository will not be added as part of this step).
- When importing a user group, its user links will be updated as per the information in the import file. In addition, if there are any users linked to the user group that are not found system, they will be added (however, the other user groups linked to that user in the LDAP repository will not be added as part of this step).
- The import will not cause any deletions of the User or User Group to occur.

### A Batch Process Initiates the Import

A batch process is used to initiate the import of information from the LDAP repository. F1–LDAP may be submitted ad hoc or may be set up in a scheduler to periodically re-sync the information from the LDAP repository into the application.

The batch process uses parameters to define how to connect to the LDAP repository. In addition, parameters are used to indicate which user or group is being imported.


Copyright © 2026, Oracle and/or its affiliates.

### Adjusting Data to Import

The system provides several mechanisms for adjusting data that is being added to the system:

- There is an **LDAP Import Preprocess** algorithm plug-in spot on the installation record. Algorithms plugged in here are called by the batch process prior to the add or update of any records. It may be used to make adjustments to the data before doing updates in the application.
- Specifically for creating or updating Users, the `F1-IDMUser` business object is used to add and create the user. The standard BO Preprocessing algorithm plug-in spot may be used to adjust data prior to creation. Note that the system does not support defining a characteristic to support a template user that can aid in populating configuration for a new user. However, this integration can take advantage of using a default template user as an algorithm parameter. Refer to Template User for more information.
- The LDAP mapping file supports some attributes to perform simple modifications to data.
  - The **transform** attribute supports values to truncate values or to convert data to upper case.
  - The **autoGenerate** attribute is specific to the User ID field. Setting this to true will trigger code that will automatically populate the User ID based on the user's name. Refer to LDAP Mapping for more information.

### Performing Additional Processing After Import

The system provides a plug-in spot on the installation record called **LDAP Import**. Algorithms plugged into this spot are called after users or user groups have been added or updated. It may be used to perform any extra processing that may need to be executed.

## Configuring LDAP Integration

To interface the LDAP based security repository with the authorization component of the Oracle Utilities Application Framework product the following must be performed:

- The location and port number of the LDAP based security repository must be defined to in the JNDI Server.
- • The LDAP based security repository must be mapped to the Oracle Utilities Application Framework security model. This mapping is expressed as an XML file containing the LDAP query, rules and defaults used in the transformation.
- • The mapping file must be configured on the `F1-LDAP` batch job.

### Define the JNDI Server

The first step in the configuration process is to define the location of the LDAP based security repository server so that the interface can connect to the physical attributes of the interface. This is done by creating a JNDI Server.

> **Note**
>
> The LDAP server is strictly not a JDNI source but is treated as a JNDI source for the integration.

Enter a reasonable JNDI Server name and description.


Copyright © 2026, Oracle and/or its affiliates.

Populate the **Provider URL** using the format `ldap://<hostname>:<portnumber>` where `<hostname>` is the host of the LDAP server and `<portnumber>` is the port used for the interface.

For the **Initial Context Factory**, the interface uses the standard `com.sun.jndi.ldap.LdapCtxFactory` provided with java for the LDAP interface. If your vendor supplies a custom context factory it may be used. Refer to the documentation provided with your LDAP based security repository for further information.

#### Define Mapping

The critical component of the interface is a file that describes the mapping between the LDAP based security repository and the system's security model. This file contains the mapping, rules and queries used by the LDAP batch program to provide the interface. The LDAP batch job includes the reference to the mapping file as a parameter. Refer to LDAP Mapping for more information on defining the mapping file.

#### Configure LDAP Batch Process

At this point, many parameters for the `F1-LDAP` batch control can be updated with system wide configuration.

- JDNI Server, User and Password may all be configured appropriately. Note that it is recommended that the **Security** setting for the Password be set to **Encrypt**.
- The **LDAP Configuration File** should be populated with the name and location of the LDAP Mapping file.
- If the LDAP service has any limitation to the number of objects that may be imported, configure the **LDAP Query Page Size** parameter to enable querying.

> **Note**
>
> **Group and User Parameters.** The assumption is that the Group or User input parameters are specific to a given import request and as such would not be populated as part of a configuration step.

> **Note**
>
> **L2 Cache**. The LDAP Import batch process requires the L2 Cache to be disabled since it needs to perform some updates in the outside of the worker threads. Any environment using LDAP Import must set `spl.runtime.batch.L2CacheMode=OFF` in the `threadpoolworker.properties` file. It is recommended to run the LDAP import in its own dedicated threadpoolworker.

- LDAP Mapping

## LDAP Mapping

An LDAP repository consists of multiple entries. Each entry represents an object in the directory that is identified by a Distinguished Name (DN) and may contain one or more attributes. In a typical LDAP repository there is usually an entry for users and an entry for groups. The connection between users and groups may be implemented in two different ways:


Copyright © 2026, Oracle and/or its affiliates.

- The users belonging to a group are defined in a special multiple-value attribute on the Group entry.
- The groups to which a user belongs are defined in a special multiple-value attribute on the User entry.

The mapping between LDAP security objects and base security objects is stored in an XML document that can be processed by the LDAP import batch job. As part of setting up your system for LDAP import, you need to define this mapping. The base package provides a sample mapping file called `ldapdef.xml` that can be used as a starting point and changed per your business requirements and your particular LDAP repository.

Once you have defined the mapping XML document, this is configured as a parameter in the `F1-LDAP` batch job.

The XML structure:

- The **LDAPEntry** element maps the LDAP entries to system objects (User or Group). The mapping file must contain one and only one LDAPEntry element for User and one for Group.
- The **LDAPCDXAttrMapping** element within the LDAPEntry element maps attributes in the LDAP entry to attributes in the system object.
- The **LDAPEntryLinks** element describes objects linked to the LDAP entry. When mapping the user entity you need to describe how the groups the user belongs to are retrieved. When mapping the group entity you need to describe how the users contained in the group are retrieved.

The following table describes the attributes to define for each element.

| Element | Attribute | Description |
|---|---|---|
| LDAPEntry | name | The name of the LDAP entry:<br>- **Group**<br>- **User** |
| | baseDN | The base distinguished name in LDAP for this entry. |
| | cdxEntity | The name of the base product entity to which the LDAP entry is mapped:<br>- **Group**<br>- **User** |
| | searchFilter | An LDAP search filter that is used to locate LDAP entries. A %searchParm% string in that filter is replaced by the value from the user or group parameter from the `F1-LDAP` batch job submission. |
| | Scope | Sets the scope of the search. Valid values are:<br>- onelevel (the value normally used)<br>- subtree |
| LDAPCDXAttrMapping | ldapAttr | The name of the LDAP attribute to be mapped. Note that this may be referenced more than once to allow one LDAP element to map to multiple base product elements. For example, if an email address should be used both for the Login ID and the Email Address. |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Attribute | Description |
|---|---|---|
| | cdxName | The name of the base product attribute to be mapped.<br>For User, this is the element within the F1-IDMUser business object.<br>For Group, this is either the ‘group’ or the ‘description’. |
| | default | The default value that will be assigned to the element referenced in the cdxName attribute when one of the following occurs:<br>- The LDAP attribute contains a null or empty value<br>- The LDAP attribute does not exist or is not specified.<br>Default values are applied only when creating a new entity and are not applied to updated entities. |
| | autoGenerate | Set this to **true** in order to turn on auto generation of the user ID. If this is true, the system will define the user id as <first initial of first name>+<last name> all uppercase, to a maximum of 8 digits. If an existing user is found for the generated ID, a number will replace the eight digit (or be appended to the end). The system will increment the number until a unique ID is found. |
| | transform | Use this attribute to indicate if a transformation of the data should occur. Valid values: `uppercase\|truncate`. Note that this attribute should not be used in conjunction with the autoGenerate attribute. |
| LDAPEntryLink | linkedToLDAPEntity | The name of the linked entity (**User** or **Group**). Use User when describing the Group entity. Use Group when describing the User entity. |
| | linkingLDAPAttr | The multiple-value attribute name on the LDAP entity that contains the linked entity. |
| | linkingSearchFilter | The search filter to be applied to retrieve the list of linked objects, for example:<br>(&(objectClass=group)(memberOf=%attr%))<br>The search filter may contain the string % attr % that acts as a substitution string and is replaced at run time by the value of the attribute named "attr" of the imported entity. If the LDAP entry you are describing is a Group and the string is %name%, it is replaced by the value of the "name" attribute of the group you are importing. If the LDAP entry you are describing is a User and the string is %dn%, it is replaced by the "dn" attribute of the User you are importing. |
| | linkingSearchScope | Sets the scope of the search. Valid values are:<br>- **onelevel** (the value normally used)<br>- **subtree** |


Copyright © 2026, Oracle and/or its affiliates.

### Sample Mapping

The following XML describes a sample mapping. The example makes the following assumptions:

- The base product attribute **displayProfileCode**is defaulted to "NORTHAM" when adding a new user.
- The LDAP Group entry contains the list of users belonging to the group in the **departmentNumber** attribute.
- The groups to which a user belongs are retrieved by applying a search filter.

```
<LDAPEntries>
  <LDAPEntry name=" User" baseDN="ou=people,dc=example,dc=com" cdxEntity="
   user" searchFilter=" (&amp;(objectClass=inetOrgPerson)(uid=%searchParm%))">
    <LDAPCDXAttrMappings>
      <LDAPCDXAttrMapping ldapAttr="uid" cdxName=" user" />
      <LDAPCDXAttrMapping ldapAttr="cn" cdxName="externalUserId" />
      <LDAPCDXAttrMapping cdxName="language" default=" ENG" />
      <LDAPCDXAttrMapping ldapAttr="givenName" cdxName="firstName"/>
      <LDAPCDXAttrMapping ldapAttr="sn" cdxName= "lastName"/>
      <LDAPCDXAttrMapping cdxName="displayProfileCode" default="NORTHAM" />
      <LDAPCDXAttrMapping cdxName="toDoEntriesAge1" default="30" />
      <LDAPCDXAttrMapping cdxName="toDoEntriesAge2" default="90" />
      <LDAPCDXAttrMapping cdxName="userEnable" default="ENBL"/>
    </LDAPCDXAttrMappings>
    <LDAPEntryLinks>
      <LDAPEntryLink linkedToLDAPEntity="Group"
linkingLDAPAttr="departmentNumber" />
    </LDAPEntryLinks>
 </LDAPEntry>
  <LDAPEntry name="Group" baseDN="ou=people,dc=example,dc=com" cdxEntity="
   Group" searchFilter=" (&amp;(objectClass=organizationalUnit)
(ou=%searchParm%))">
    <LDAPCDXAttrMappings>
      <LDAPCDXAttrMapping ldapAttr="name" cdxName="Group" />
      <LDAPCDXAttrMapping ldapAttr="description" cdxName=" Description"
default="Unknown" />
    </LDAPCDXAttrMappings>
    <LDAPEntryLinks>
      <LDAPEntryLink linkedToLDAPEntity="User" linkingSearchFilter="
      (&amp;(objectClass=inetOrgPerson)
(departmentNumber=%distinguishedName%))"
     linkingSearchScope="onelevel" />
    </LDAPEntryLinks>
  </LDAPEntry>
</LDAPEntries>
```

# Oracle Identity Manager Integration

The *Oracle Identity Manager* product allows a site to centralize their user definitions and password rules to manage and deploy across the enterprise set of products. When an employee joins an organization, changes their name or departs an organization their security


Copyright © 2026, Oracle and/or its affiliates.

presence across an enterprise must be appropriately managed. Oracle Identity Manager allows for users to be provision and managed in a central location.

An integration is provided to allow the ability to create, maintain and remove users in the identity management product and sync those changes to the users defined in the application. The following sections provide additional details about the integration with respect to configuration steps required in an Oracle Utilities Application Framework based product. For more information about the configuration required in the identity management product, refer to the *Identity Management Suite Integration* technical reference paper.

In order to use this functionality, feature configuration options for the **External Messages** feature type must be configured.

- Set option type **Support SPML Deployment in IWS** to **true**.
- Set option type **Default SPML service security policy** to an appropriate value per your implementation rules.

### Template User Functionality

The user object in this product captures configuration used to control access but also preferences. The identify management product allows for extending the configuration to capture user configuration that is specific to this product. However, it does not support providing searches or dropdowns to select valid values. For example, to define the user's Home Page requires the reference to a navigation option. To set up your business process such that the home page is configured when defining the user in the identity management product dictates that the security user types in the correct navigation option reference.

On the other hand, to define a minimal amount of user information in the identity management product may result in a two step process for defining users: first define them in the identity management product with the basic authentication details and setting system defaults for some important fields, then after submitting the new user to be added to this product, navigate to the user page in this product and fill in all the configuration that is specific to this product.

The product provides support for defining a template that can facilitate the definition of users and reduce some of the challenges listed above. Refer to Template User for information about defining template users along with other details around this functionality.

Once you have template users defined, you should look at extending the configuration in the identity manager product.

- Map the information that is unique to a user and in addition, define a field for the template user. For example, you may choose to only capture the Name (first and last), Email address and User IDs for the user along with its Template User (which is mapped to a user characteristic). Additional fields may be included for capture in the identity management product when defining new users as per an implementation's business needs. For example, if the organization covers multiple time zones, perhaps it is easier to define the user's time zone when defining the user in the identity management product.
- When the new user is uploaded to the system, the algorithm used by the system to create the user looks for the existence of a template user sent as a characteristic of type `F1-TMUSR`.
- Once the new user is created, its configuration can now be adjusted, if applicable.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> There is configuration needed in Oracle Identity Management to capture the template user and any other information that the implementation has chosen to define in the identity management product when provisioning a new user. Refer to the *Identity Management Suite Integration* technical reference paper for more information.

# Provision Users with SCIM Open Standard

The product provides a REST web service `F1-SCIMUser` (SCIM User Provisioning) that supports adding, changing or viewing details of a user where the API follows the SCIM standard.

The following points highlight some of the mapping between the SCIM API and the application's user record.

- The user record in the application supports only first name and last name. The SCIM standard supports additional detail such as middle name, suffix and prefix. These elements are ignored when adding or updating a user.
- The SCIM standard supports a collection of email addresses. The application only supports one email address. As such, only the first email address is used when adding or updating a user.
- The integration relies on the use of a Template User so that the user is created with appropriate application configuration associated with each user. The system copies application settings from the template user to the new user being provisioned. If the external system is able to support referencing a template user such that it can be provided with the payload, the integration expects the user type element in the API to reference the template user. If it's not possible to provide a template user ID in the user type element, you should define a default template user using an algorithm parameter. Refer to Template User for more information.

Refer to the web service definition for more information about the supported integration.

### Template User Functionality

The user object in this product captures configuration used to control access but also preferences. The identify management product allows for extending the configuration to capture user configuration that is specific to this product. However, it does not support providing searches or dropdowns to select valid values. For example, to define the user's Home Page requires the reference to a navigation option. To set up your business process such that the home page is configured when defining the user in the identity management product dictates that the security user types in the correct navigation option reference.

On the other hand, to define a minimal amount of user information in the identity management product may result in a two step process for defining users: first define them in the identity management product with the basic authentication details and setting system defaults for some important fields, then after submitting the new user to be added to this product, navigate to the user page in this product and fill in all the configuration that is specific to this product.

The product provides support for defining a template that can facilitate the definition of users and reduce some of the challenges listed above. Refer to Template User for information about defining template users along with other details around this functionality.

Once you have template users defined, you should look at extending the configuration in the identity manager product.


Copyright © 2026, Oracle and/or its affiliates.

- Map the information that is unique to a user and in addition, define a field for the template user. For example, you may choose to only capture the Name (first and last), Email address and User IDs for the user along with its Template User (which is mapped to a user characteristic). Additional fields may be included for capture in the identity management product when defining new users as per an implementation's business needs. For example, if the organization covers multiple time zones, perhaps it is easier to define the user's time zone when defining the user in the identity management product.
- When the new user is uploaded to the system, the algorithm used by the system to create the user looks for the existence of a template user sent as a characteristic of type `F1-TMUSR`.
- Once the new user is created, its configuration can now be adjusted, if applicable.

> **Note**
>
> There is configuration needed in Oracle Identity Management to capture the template user and any other information that the implementation has chosen to define in the identity management product when provisioning a new user. Refer to the *Identity Management Suite Integration* technical reference paper for more information.

# User Redirection

The product allows an implementation to support the ability for a customer to authorize a customer service representative (CSR) to view the customer's account information in a customer facing portal (Digital Self Service), to see exactly what the end customer is seeing/ experiencing.

When the request for the CSR to view the customer information is received, the following steps occur:

- The security server for the customer facing portal is contacted to obtain a temporary grant code.
- The redirection response returns the grant code and the URL of the target customer facing portal.
- The browser redirects to the user to this URL with the grant code, which is further authenticated and automatically exchanged for an authorization token, allowing the CSR to log in.

Standard security measures with private/public key encryption using JSON Web Tokens (JWT) ensures that only the implementation's enterprise application is allowed to communicate with the implementation's customer facing portal server. The system also provides a plug-in spot to allow for additional information, such as the customer's name and email address, to be included in the call to the security server.

This feature works in conjunction with functionality provided by the enterprise customer application. Refer to the documentation for that product for information about how this feature is supported.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The product has been designed to potentially support other types of redirection besides the mentioned use case by adding another entry in the Redirection Target lookup (F1_REDIRECTION_TARGET_FLG). However, the internally provided services work specifically for the Digital Self Service use case. There is no guarantee that they would work for a different redirection use case. Please contact product support if your implementation would like to implement a different redirection scenario.

#### Redirection Security Context Plug-in

The product supports a plug-in spot to allow for the system to populate additional keys and values to add to the JWT (JSON Web Token) when calling the external security service.

In addition, the product provides an Extendable Lookup BO **Redirection Security Context (F1-RedirectionSecurityContext)** to define the key mnemonic values. For an enterprise product or implementation to populate additional keys, use the extendable lookup to define the key mnemonics. Then provide an algorithm that uses the available information about the customer and populates the security context keys and values by determining the related data in the system.

#### Master Configuration

The master configuration **Redirection Targets Configuration (F1-RedirectionTargetsConfig)** defines the information needed to contact the security server to obtain a grant code. It also defines the redirection URL.

#### Build an Anchor Hyperlink

The service script **Return a Hyperlink to the Redirect Servlet for a given Target (F1TargetLink)** is provided to allow for a link to be placed on a UI map or in a zone to initiate the redirection request. Refer to the service script detailed description in the metadata for more information.

# Data Synchronization

Your implementation may need to communicate certain data to external systems. This may be part of a data warehousing requirement or an integration effort. The synchronization process has two main parts. First, the change to the data must be detected and captured. Once that is accomplished, the next step is to manage the communication of that change to the external systems involved. The changes must be captured in chronological order so as to avoid systems going out of sync.

- Understanding Data Synchronization
- Maintaining Sync Requests

## Understanding Data Synchronization

The product supports various options for managing an ongoing synchronization of an entity to an external system. While each integration involves unique business rules, in many cases the process is quite similar and involves the following overall steps:

- When an entity is added, changed or deleted logic determines whether the change is eligible for synchronization and if so a sync request record is initiated to manage the


Copyright © 2026, Oracle and/or its affiliates.

synchronization process. The base product uses the **Audit** plug-in spot on the maintenance object to allow for such logic to be performed when a change is detected to a record for that MO. The framework calls the algorithm defined on this plug-in spot in the event a change to the MO has been detected. Refer to the description of the plug-in spot on Maintenance Object - Algorithms for more information about when this plug-in is called.

- The processing of the sync request is deferred to batch so that the communication with the external system does not impact the updating transaction. It also ensures that the data being shared with the external system is fully committed to the database.

> **Warning**
>
> Do not remove the deferred batch contol from business object status definitions of such synchronization requests. These flows must remain deferred as designed for data integrity reasons.

- When processed, sync request logic manages the following aspects of the process:
  - Determines whether the synchronization is still needed and if not discards the request.
  - Prepares the outbound message with the entity's synchronization details and sends it to the external system.
  - Manages errors and retry attempts as needed.

The following sections describe the various data synchronization options supported by the product.

- Accumulative Data Synchronization
- Generic Data Synchronization

## Accumulative Data Synchronization

The product supports a data synchronization method by which several changes made to an entity may be accumulated and sent together as a single message to the external system.

The following points describe the functionality at a high level:

- When an entity is added, changed or deleted, a snapshot of the entity’s data from before the change is captured on one or more sync requests, one for each external system the data needs to be interfaced to. The before-update snapshot is used as a base line for detecting all changes made since then at the time of sending the message to the external system.
- Sync requests are processed by the next monitor batch process.
- The processing of the sync request typically involves business rules that compare the before-update snapshot with the current content of the entity to detect all change made. If changes were made, logic may decide to proceed with the sending the synchronization message otherwise the request is discarded. These rules are also responsible for handling any errors encountered while communicating with the external system and manage retry attempts as needed.

This functionality is supported by logic supplied in the base business object Sync Request (**F1-SyncRequest**). Note that each edge application delivers an appropriate child business object for this BO for each specific synch scenario supported in that product. Some of the functionality below is accomplished using configuration on the parent BO delivered by the framework while other functionality may be delivered by the child BO. In addition, there may be more complex use cases supported by your specific product integration. Refer to your specific application’s


Copyright © 2026, Oracle and/or its affiliates.

library of Sync Request business object along with the documentation related to your specific product integration for more information.

The following sections describe key aspects of this data synchronization approach.

### Capturing the Change

The base product provides a change data capture (MO Audit) algorithm `F1-GCHG-CDCP` that may be used by maintenance objects that need to be synchronized by this method. This algorithm creates a Sync Request record for the changed record, capturing the MO code and the primary key, if it doesn't find an existing sync request for the same record (and the same business object) in the initial state. The sync request business object used is the one defined in the **Sync Request BO** option on the MO for the record that was changed. If more than one such option exists then multiple sync requests are created.

Your specific product may also introduce additional Audit algorithms to cater for more sophisticated examples.

When creating the sync request record, typically the Sync Request BO will have a pre-processing plug-in that captures a snapshot of the record's data prior to its change. This will be used in subsequent steps to verify that the external system needs to be notified of the change.

### Confirming that a Sync is Needed

Once a sync request is captured, there are several steps performed prior to any information being sent to the external system.

> **Note**
>
> This section only highlights key steps. Please refer to the business object configuration, its lifecycle and algorithms for a thorough picture of the full functionality..

- When a Sync Request record is created, its initial state (**Pending**) is configured to be processed by a batch monitor. That way, records are added to the sync request table throughout the day but all are processed together. The MO audit algorithm ensures that a new synch request is not created if a Pending record already exists for a given MO / PK combination (for the same business object). However, it is possible that a record for that MO / PK exists in a subsequent "non-final" (such as **Awaiting Acknowledgement**). This state includes a monitor algorithm to check for that condition and to skip transitioning if another record exists. This is done to ensure that the existing record is fully processed before this new record is processed.
- The next state of the lifecycle is **Determine if Sync Needed**. This step uses an algorithm to take a snapshot (called the 'final snapshot') of the data and compare it against the initial snapshot taken when the record was created. Based on the logic of the algorithm, it may decide to proceed (transition to **Send Request** or to discontinue (transition to **Discarded**.

### Communicating to the External System

Once it is confirmed that the sync should occur, a message must be sent to the external system. The following points highlight the basic functionality.

- An algorithm linked to the **Send Request** state. The expectation is that this algorithm creates an outbound message that routes the information to the external system appropriately. The algorithm must determine the external system and outbound message type to use. Business objects for Sync Request support BO options to define the external system and outbound message type to use for this algorithm.


Copyright © 2026, Oracle and/or its affiliates.

- Once the outbound message is triggered, the record transitions to the **Awaiting Acknowledgement** status. This state is used to hold the sync request from further state transitions until an acknowledgement is received from the external system. Note that this step relies on implementation of a response mechanism from the external system. It is recommended to implement a response as this helps control the chronological flow of information. The product supplies the business service `F1-UpdateSyncRequest` that transitions the sync request to either the next default state (in this case the **Synchronized** state) if a positive acknowledgement is received; or the state associated with the Rejection transition condition (in this case the **Error** state) if a negative acknowledgement is received. In addition, this state may be configured with a monitor algorithm that detects that a timeout limit has been reached.
- For records that enter the **Error** state, it is recommended to configure an algorithm that creates a To Do entry to alert someone of the problem. Refer to the integration documentation for more information. The state is already preconfigured with an algorithm to complete To Dos when exiting the state.
- The final state **Synchronized** is used to mark the successful synchronizations. However, for more complicated use cases, this state may be used to trigger some additional action. Refer to the documentation for your specific product integration for more information.

## Generic Data Synchronization

The product supports a generic data synchronization method by which each change to an eligible entity triggers a synchronization flow that communicates details about the entity to one or more external systems.

The following points describe the functionality at a high level:

- The integration of a specific entity (maintenance object) and an external system is enabled and configured using a **Data Synchronization Control** (F1-DataSyncControl) extendable lookup record.
- Rules supporting the integration reside in a **synchronization script** referenced on the control record.
- When an entity is added, changed or deleted, a sync request may be initiated for each integration according to business rules defined by the corresponding synchronization script.
- Sync requests are processed by the next monitor batch process.

> **Note**
>
> In a cloud installation, near real-time processing is supported for time depenent synchronizations. Refer to Deferred Requests Queue for more information.

- When the sync request is processed, data is synchronized to the external system according to business rules defined by the corresponding synchronization script.

This functionality is supported by logic supplied in the base business object Generic Data Sync Request (**F1-GenericDataSync**). This business object does not need to be extended to support specific integration rules as these rules reside in the synchronization script associated with each integration's data synchronization control record.

The following sections describe key aspects of this data synchronization approach.


Copyright © 2026, Oracle and/or its affiliates.

### Data Synchronization Control

The **Data Synchronization Control** (F1-DataSyncControl) extendable lookup defines the configuration needed to support an ongoing data synchronization of an entity to an external system. The configuration defines the entity's maintenance object, the external system it is synchronized with and references a synchronization script that manages the specific synchronization rules for this integration. An integration is considered enabled when its corresponding control record is in active status.

The data synchronization script is called in the following modes:

- When the entity is added, changed or deleted, the script is called in **Check** mode to determine whether a sync request should be created for the changed entity and a specific external system.
- When the sync request is processed, the script is called in **Process** mode to prepare and send the outbound message to the external system. Alternatively the script may request to discard the request or indicate that it should be processed later. The latter may be used when there is a need to wait until certain conditions are met before sending the message to the external system.

Note that the configuration for a maintenance object and a specific external system may be overridden by another configuration defined with a higher sequence number.

### Confirming that a Sync is Needed

The base product provides a change data capture (MO Audit) algorithm `F1ONGDATASYN` that may be used by maintenance objects that need to be synchronized by this method. The algorithm consults the synchronization script associated with each active integration control record configured for the entity as to whether the external system needs to be notified for this change. Once confirmed, the algorithm creates a Sync Request record for the changed record, capturing its MO code and the primary key, along with the corresponding external system.

Note that the decision as to whether to create a sync request or not is controlled by the integration specific synchronization script while the creation of the sync request is the responsibility of the audit algorithm. As such, the same sync request BO is used for all integrations, with no provision for capturing integration specific details.

It is recommended to design the synchronization script to only approve the creation of a synchronization request when it is needed. This prevents unnecessary creation of sync requests that would otherwise be discarded when eventually processed.

### Managing the Synchronization Process

Once a sync request is captured, there are several steps performed prior to any information being sent to the external system.

> **Note**
>
> This section only highlights key steps. Please refer to the business object configuration, its lifecycle and algorithms for a thorough picture of the full functionality..

- When a Sync Request record is created, its initial state (**Pending**) is configured to be processed by a batch monitor. That way, the changes that triggered the request are fully committed before being communicated to an external system.


Copyright © 2026, Oracle and/or its affiliates.

- When the sync request record is processed, a monitor algorithm calls the synchronization script associated with the corresponding integration's control record to manage the synchronization process. The synchronization script may indicate either of the following:
  - The request is not yet ready to be processed, i.e. it should remain in its current status and get processed again by the next batch run.
  - The request should be canceled. This indication results in the transition of the record to the **Canceled** status.
  - The request was successfully processed. The synchronization script has prepared the outbound message and routed the information to the external system appropriately. This indication results in the transition of the record to the **Sent** status.
- If an error occurs while processing the request, a To Do Entry is raised to capture the error details and the record is set to the **Error** status. Retry processing of the sync request continues with each batch process until the request is either sent or canceled.

#### Communicating to the External System

Typically the communication with an external system is finalized when the message is successfully sent and successfully processed by the external system. Depending on the specific integration, the message may not be processed right away by the external system but rather queued and processed later. In this situation, the sync request may need to be further acknowledged as successfully processed (set to the **Acknowledged** status) upon an explicit response mechanism from the external system. The product supplies the web service `F1-UpdateAndTransitionSyncRequest` that may be used to process such acknowledgements.

## Maintaining Sync Requests

The system provides a Sync Request portal that is used to view the in progress or completed sync request records.

The menu location of the portal depends on your specific edge product. It may be in a Data Synchronization menu or perhaps in the Batch menu. You are brought to a query portal with options for searching. The options may differ based on your specific product.

Once a sync request has been selected, you are brought to the maintenance portal to view and maintain the selected record.

An **Actions** zone may appear to display specific actions. Alternatively, the actions may be displayed directly in the display area of the **Sync Request** zone.

The **Sync Request** zone provides basic information about the sync request record.

Depending on your specific product additional zones may appear.

# Operational Analytics

The framework provides several building blocks and tools that the edge applications may use to implement analytic reports directly from the operational application using Oracle Utilities Analytics Visualization product (referred to in this section as the analytics visualization product). The following sections provide more information about this functionality.

- Understanding Operational Analytics
- Calendar and Time Dimensions
- Maintaining Bucket Configurations


Copyright © 2026, Oracle and/or its affiliates.

- Defining Analytics Options
- Defining Analytics Tables
- Defining Analytics Attribute Mapping

# Understanding Operational Analytics

The following sections describe the type of configuration supported in your product to integrate with the analytics visualization product. Refer to the Oracle Utilities Analytics Visualization documentation for more information.

### Direct Data Access

The analytics visualization product's canvases report directly off of the operational system (with the exception of batch run analytics) thus eliminating the need of an ETL process. Selected tables and views have been designated as dimensions and facts for the purpose of generating the Start Schemas used by these canvases.

### Batch Run Analytics

There is a high volume of batch run data in the operational system that changes frequently and is recorded in a complex set of relationships. The product provides snapshot tables to capture only the batch run and thread data required for analytics. Using these tables for batch subject area views both improves performance and supports extracting the data for upload to other environments.

The snapshot tables are populated in increments. The **Batch Run Analytics Snapshot Update** and the **Batch Thread Analytics Snapshot Update** processes are run on an ongoing basis; each snapshot update selects only the batch runs and threads not yet included in the snapshot. Parameters are supplied to restrict the selected data based on a range of days from the current date, to support the initial uptake of data. It is recommended that batch run data is updated before the batch thread data, as only threads with corresponding records in the batch run snapshot will be added. Refer to batch controls `F1BRANSN` and `F1BTANSN` for more information.

The **Batch Run Analytics Data Export** and **Batch Thread Analytics Data Export** processes are provided to extract the batch data in increments for upload to other environments. These processes assume that a **Data Export Control** exists for each of the snapshot maintenance objects, to capture the date of the previous extract as a reference point for selecting the next set of records. Refer to batch controls `F1BRANEX` and `F1BTANEX` for more information.

### Calendar and Time Dimensions

Analytic reports rely on calendar and time dimensions to support various hierarchical grouping by date and time.

The Calendar dimension provides a level-based definition of the standard calendar and fiscal calendar in a flattened representation that is commonly used in data warehouses. This is necessary to enabling customers to group by calendar week, months, quarter and fiscal calendar period, quarter and fiscal year etc. Note that fiscal information about a specific date is optional and sourced from your edge application specific accounting calendar

In the same way, the Time dimension provides a level-based definition of each minute in a day, supporting reports that group by hours, AM/PM etc.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Calendar dimension records are generated by a batch process; one record for each date in a specified period of time. You need to run the **Generate Calendar Dimension** batch process on an ongoing basis to cover the standard and accounting calendar days needed to support your business. Refer to `F1-BICDD` batch control for more information.

> **Note**
>
> Time dimension records are also generated by a batch process; one record for each minute in a day. You need to run the **Generate Time Dimension** batch process once to generate all the records. Refer to `F1-BITMD` batch control for more information.

### Bucket Configuration

The analytics visualization product provides support for defining a set of ranges, each representing a bucket for which extracted measures can be grouped and classified under the relevant bucket. The framework product provides support for viewing and defining the buckets. Refer to Bucket Configuration for more information.

### Analytics Attribute Mapping

The product supports mapping of characteristics to user defined fields associated with dimensions in the analytics visualization product. Each characteristic table that is associated with a dimension table is provided with pre-generated user defined fields identified by unique column sequence numbers in the analytics visualization product. The mapping of characteristics to dimension user defined fields is maintained directly on the dimension portal. You may also review these mappings using the Analytics Attribute Mapping portal.

### Logical Dimension

The Element Name of a foreign key field to a dimension represents a distinct logical version of the dimension. This allows for different logical dimensions to be based on the same dimension table. For example, a "Location" dimension may be referenced as a "Main Location" on one fact and also as an "Alternate Location" on the same or other fact. While both elements refer to the same dimension as foreign keys, each represent a distinct logical dimension via their element names. In the same way, while each date field is implicitly considered as a foreign key to the Calendar dimension, each distinct element name establishes a unique logical dimension. For example, a "Start Date" element name and an "End Date" element name are distinct logical dimensions of the underlying Calendar dimension.

Some logical dimensions may represent an aggregated or summarized form of another logical dimension. They are also known as "shrunken" dimensions. For example, the "Calendar Month Dimension" view is an aggregated dimension of the Calendar dimension. Establishing a link between a summarized dimension and its detailed dimension allows the analytics visualization tool to drill from one fact to another for performance reasons when they are indirectly linked via the same logical dimension but at different aggregation levels. For example, "Fact A" represents daily records and "Fact B" aggregates the same measures at a monthly level. Associating the "month" field in "Fact B", which is a foreign key to the "Calendar Month Dimension" view, with the logical dimension name of the "date" field in "Fact A" allows analytic queries to use "Fact B" for queries at the monthly level or above which perform better than performing them on the more granular "Fact A".


Copyright © 2026, Oracle and/or its affiliates.

### Analytics Configuration

Depending on which product or products you have installed, there may be some configuration needed for the analytics visualization product. Refer to Defining Analytics Options for more information.

## Calendar and Time Dimensions

This portal is used to view generated calendar and time dimension records.

Refer to Understanding Operational Analytics for an overview of operational analytic functionality.

Navigate using **Admin**, and then **Analytics**, and then **Calendar and Time Dimensions**.

The following zones may appear as part of the portal's **Main** tab page:

- **Calendar Dimension.** This zone lists most recent calendar dimension records. You may filter the list to review records generated up to a specific date.
- **Calendar Date.** This zone provides information about a selected calendar date.
- **Time Dimension.** This zone lists all time dimension records.

## Maintaining Bucket Configurations

Several key performance indicators in the analytics product look at measurement values (for example: the age of an asset or the age of debt) classified into a number of pre-defined groups also known as buckets. The overall metric can then be reported by the different buckets and allow various analyses.

For example, the age of an asset can be classified into the following buckets:

- Less than 6 Months
- 6-12 Months
- One Year and Older

The age of debt, also known as arrears can be classified onto the following buckets:

- 0-30 Days
- 30-60 Days
- 60-90 Days
- 90+ Days

The definition of the buckets is extracted to the Business Intelligence data warehouse, to be used as dimensions.

### Bucket Definition Considerations

Each type of bucket is defined using a bucket configuration Business Object. The bucket definition considerations and/or rules will vary based on the bucket configuration business object used. The business objects available are driven by your specific product. For a list of available bucket configurations business objects, navigate to the business object page and view the business objects for the Bucket Configuration maintenance object.


Copyright © 2026, Oracle and/or its affiliates.

### Setting Up Bucket Configurations

To maintain the bucket ranges for the bucket configuration(s) applicable to your product, open **Admin**, and then **Analytics Configuration**, and then **Bucket Configuration.**

You are taken to the query portal where you can search for an existing bucket configuration. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

> **Note**
>
> Your specific product may also include an Analytics Configuration portal that displays the list of existing and potential bucket configuration records, allowing you to drill into this page to view the record in detail.

The **Bucket Configuration** zone provides basic information about the bucket configuration.

For more information about the elements supported refer to the zone help or to the relevant analytics integration documentation for your product.

## Defining Analytics Options

Refer to Understanding Operational Analytics for an overview of the operational analytics functionality.

Your specific product may require some system wide configuration to be defined when using operational analytics.

Navigate to the Extendable Lookup portal. Search for and select the **Analytics Options** business object.

Use the embedded help to review information about the options to define.

## Defining Analytics Tables

This portal is used to search for Analytics Table definitions. An Analytics Table record exists for each table or view in the operational system that represent dimension and fact tables in the Operational Analytics product.

Refer to Understanding Operational Analytics for more information.

You can access the portal from the**Admin Menu**, and then **Analytics Configuration**, and then **Analytics Table.** You are brought to a query portal with options for searching for a specific record. Once a record has been selected you are brought to its maintenance portal to view and maintain the selected record.

- Defining Analytics Dimensions
- Defining Analytics Facts

## Defining Analytics Dimensions


Copyright © 2026, Oracle and/or its affiliates.

This portal is used to maintain metadata for Analytics Dimension tables. A dimension record exists for each table in the operational system that represents a dimension table in the Operational Analytics product.

Refer to Understanding Operational Analytics for more information.

You can access the portal from the**Admin Menu**, and then **Analytics Configuration**, and then **Analytics Table.** You are brought to a query portal with options for searching for a specific record. Once a dimension record has been selected you are brought to this portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Analytics Table.** This zone provides information about the selected dimension record.
- **Fields.** This zone lists all the fields that define the dimension table. Fields may be selected from the dimension's underlying operation table using the **Excluded Fields** zone or added as computed fields.
- **Dimensions.** This zone lists dimension tables that this dimension references via foreign key constraints.
- **Referenced By.** This zone lists dimension and fact tables that reference this dimension via foreign key constraints.
- **Characteristic Mapping.** This zone manages characteristic mapping extensions for the dimension. Refer to analytics attribute mapping for more information.
- **Excluded Fields.** This zone lists fields from the dimension's operational table that are not included in the dimension table.

> **Note**
>
> The ability to introduce custom analytics dimensions is currently not supported.

## Defining Analytics Facts

This portal is used to maintain metadata for Analytics Fact tables. A fact record exists for each table or view in the operational system that represents a dimension table in the Operational Analytics product.

Refer to Understanding Operational Analytics for more information.

You can access the portal from the**Admin Menu**, and then **Analytics Configuration**, and then **Analytics Table.** You are brought to a query portal with options for searching for a specific record. Once a fact record has been selected you are brought to this portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Analytics Table.** This zone provides information about the selected fact record.
- **Fields.** This zone lists all the fields that define the fact table. Fields may be selected from the fact's underlying operation table or view using the **Excluded Fields** zone or added as computed fields.
- **Dimensions.** This zone lists dimension tables that this fact references via foreign key constraints.


Copyright © 2026, Oracle and/or its affiliates.

- **Excluded Fields.** This zone lists fields from the fact's operational table or view that are not included in the fact table.

> **Note**
>
> The ability to introduce custom analytics facts is currently not supported.

## Defining Analytics Attribute Mapping

This portal maps extensions to user defined fields provided for tables associated with an analytics dimension table that capture well defined extension data. A common example of this type of data are characteritics.

These mappings may be directly maintained on the operational Analytics Dimension portal. Refer to Understanding Operational Analytics for more information.

If your system is integrated with the Oracle Utilities Analytics product refer to Understanding Analytics Integration for more information about ETL based characteristic mapping.

You can access the portal from the **Admin Menu**, and then **Analytics Configuration**, and then **Analytics Attribute Mapping.** You are brought to a query portal with options for searching for a specific mapping record. Once a record has been selected you are brought to the this portal to review the information for the selected record.

The **Analytics Attribute Mapping** zone on the portal's **Main** tab page provides information about the mapping record.

> **Note**
>
> For characteristic mapping, only **Adhoc** and `Pre-defined` characteristic types are supported.

> **Note**
>
> For the product provides support for identifier mapping for operational analytics in this release but only the metadata definition is supported. The support for these types of attributes in the operational analytics product is in a future release.

# Analytics Integration

The framework provides several building blocks and tools that the edge applications may use to implement integration with the Oracle Utilities Analytics product (referred to in this section as the analytics product). The following sections provide more information about this functionality.

- Understanding Analytics Integration

## Understanding Analytics Integration

The following sections describe the type of configuration supported in your product to integrate with the analytics product. Refer to the Oracle Utilities Analytics documentation for more information.


Copyright © 2026, Oracle and/or its affiliates.

### Master Configuration

Edge applications that include an integration to the analytics product typically include a master configuration record that captures information needed for the extract, load and transformation step, such as extract parameters. These records are provided by the specific edge products and my be viewed and maintained on the master configuration portal.

Note that your specific edge application may deliver an **Analytics Configuration** portal that displays the information from the master configuration record along with other analytics related configuration.

### Bucket Configuration

The analytics product provides support for defining a set of ranges, each representing a bucket for which extracted measures can be grouped and classified under the relevant bucket. The framework product provides support for viewing and defining the buckets. Refer to Bucket Configuration for more information.

### ETL Based Characteristic Mapping

The product provides objects to allow mapping configuration of characteristics in the product to user defined fields on dimensions in the analytics product. The mapping is used by the extract process to populate target user defined columns with corresponding characteristic data.

This type of mapping relies on configuration in the Allowed Target Dimensions extendable lookup. The extendable lookup is used to define each Target Table in the analytics product that has one or more user defined fields that may be populated with characteristic values. It also defines the valid characteristic entities that may act as the source for the characteristic data.

> **Note**
>
> The framework does not provide any values in this lookup, but edge products that support mapping provide values in this lookup. Please refer to your specific product's integration chapter and refer to the Oracle Utilities Analytics documentation for more information.

Refer to Defining Analytics Attribute Mapping for more information.

### Change Data Capture Using Sync Request

Depending on the specific edge application and version you are using, there may be components of the integration that use Sync Request for the change data capture step. If that functionality applies to your implementation, the following points highlight how to get more information:

- Refer to the administration guide for Oracle Utilities Analytics to confirm if your product integration is using Sync Request for any change data capture functionality.
- Review the Sync Request Business Objects provided by your product for analytics integration.
- Refer to Data Synchronization for a high level understanding of the process.


Copyright © 2026, Oracle and/or its affiliates.

# Business Flags

It is possible that information detected in one product may be useful or even critical to share with another product. The framework provides functionality for receiving information from an external system that acts as a type of flag or alert that may need investigation. This allows any system to store detected business flags in a common way and share that information with one or more other systems.

- Understanding Business Flags
- Setting Up Business Flag Configuration
- Maintaining Business Flags

## Understanding Business Flags

The following is an example of a use case for business flags. Imagine that DataRaker highlights potential theft of service at a certain location. That product may initiate a business flag alert to various products owned by the implementation with a recognized "standard name" for the business flag, such as "TAMPER".

- If Oracle Utilities Meter Data Management receives this business flag, it may initiate a service investigation monitor.
- If Oracle Utilities Mobile Workforce Management receives this business flag, it may initiate a service investigation activity.
- If Oracle Utilities Customer Care and Billing receives this business flag, it may initiate a hold on billing for that location.

Note that the framework product supplies basic functionality to support logic that is common to all edge applications that implement business flag functionality. However it is the individual edge applications that supply more specific functionality (business objects and algorithms) for specific use cases, if applicable.

The following sections highlight functionality supported for business flags in the framework. Refer to the edge application product documentation for more details for supported use cases.

- Standard Name
- Business Flag Type Defines Behavior for a Standard Name
- Business Flag Type Algorithms
- Objects Linked to a Business Flag
- Impacted Business Process
- Dates
- Creating Business Flags
- Confidence

### Standard Name

To ensure that Business Flags are universally understood across all edge applications and to simplify integration each Business Flag will have a Standard Name. This is a name that is used by all the products when sending information to each other. That way, if DataRaker product sends Oracle Utilities Meter Data Management a "TAMPER" business flag, it should result in


Copyright © 2026, Oracle and/or its affiliates.

the same functionality as when Utilities Customer Care and Billing sends a "TAMPER" business flag.

Standard Names are defined using an extendable lookup. In addition to common extendable lookup fields, the standard name also references a category and a class. The lookup also supports defining one or more external names, for cases where information is communicated from an external system that does not send the expected Standard Name.

The framework delivers the class values but does not deliver any standard names or category values. Refer to your specific edge application products to verify if any standard names or categories are delivered. Your implementation may configure appropriate standard names and categories based on your business rules.

## Business Flag Type Defines Behavior for a Standard Name

Although the definition of the business flag standard names should be universally understood by the various integrated products that support them, each individual product defines what should occur when a business flag with a given standard name is created. This is configured using a business flag type. Only one active business flag type may exist for a given standard name. Business flags that are received from an external system will define the standard name, but will not have knowledge of the specific business flag types defined. The business flag type is determined based on the standard name.

The business flag type defines the business object to use when creating the resulting business flag record. The business object defines the lifecycle of a resulting business flag record.

## Business Flag Type Algorithms

The business flag type includes support for algorithms. This allows for an implementation to define a common business object that may be used for different business flag types (if a common lifecycle is followed) but allow for different functionality to kick in depending on the business flag type.

The product supplies a plug-in spot for **Additional Processing** that may be invoked by a business flag that enters the Additional Processing state. Refer to your product's library of business objects to determine if there is an Additional Processing state that supports calling algorithms on the business flag type.

## Objects Linked to a Business Flag

There are two types of links between an object in the system and a given business flag.

**Affected Entity**

Each business flag is associated with a single record in the system that is considered the "affected entity" or the entity that the business flag is associated with. The affected entity is defined by the specific business objects designed for the use cases supported by your edge product. For example, many utility base products may configure service point as the affected entity for its business flag use cases. Each business flag created is linked to a specific service point. Linking a specific entity to each business flag allows for algorithms to trigger functionality for that entity such as an investigation order. In addition, algorithms may be implemented in other business process areas that look for the existence of a business flag and act accordingly.

**Related Objects**

The business flag supports linking one or more related objects to a business flag to make it easier to trigger functionality or for impacted business processes to look for business flags. For


Copyright © 2026, Oracle and/or its affiliates.

example, when creating a business flag for a given service point, it may be useful to link all the accounts that are currently linked to the service point. Then, if an account oriented process should check for a business flag, it can look directly for a business flag linked to the account in its related object.

## Impacted Business Process

The product supplies support for associating one or more impacted business processes to a business flag type. This configuration is used when functionality for that business process is impacted in some way based on the existence of a business flag of a given type. For example, maybe some process is put on hold when a certain type of business flag exists.

Note that configuring a business process on business flag type is not enough to trigger any impact on that business process when a business flag exists. There must also be some logic implemented in the business process functionality itself that knows to look for a business flag for a given record that is configured to impact the business process.

The definition of the business process is at the discretion of the edge application that supplies functionality to support this. For example, the business process could be defined as something broad such as "billing" or could be something more granular such as "billing estimation". The system supplies an extendable lookup to use for configuring the supported business processes. Refer to the values of the business process extendable lookup in your edge application or to the edge application specific Business Flag documentation for more information about supported business processes.

## Dates

A business flag supports two dates: a Business Flag Date/Time and a Business Flag **End** Date/Time

- The business flag date / time is required for all business flags. For some types of business flags only one date is needed.
- For business flags that have a start and end period, the business flag date/time acts as the start date and the other field is the end date.

For a business flag that has a date range, it may be important for functionality implemented for impacted business processes. How the process treats the date will depend on its functionality.

- For some processes, the business flag is essentially expired after the end date has passed. This applies to impacted processes that are only looking at the current status of data in the system. For example, collection processing could be held if there is a business flag currently in effect (where the current date is within the date range). It would never look at historical business flags.
- For some processes where historical data may be relevant, a business flag effective during that same historical period may impact the process. For example if a business flag denotes an outage event for a given time period, perhaps estimated consumption should never be calculated for that time period.

Note that because business flags have a status, the design for the lifecycle of the business objects for the above effective dated use cases must carefully consider the states. For business flags that are considered expired after the end date passes, the BO lifecycle may be designed to transition to a final state after that date such that the record is no longer included in active processing. For business flags that continue to impact processing for a historic period, the BO lifecycle may be designed to remain in a non-final state such that it is clear that the record is still applicable.


Copyright © 2026, Oracle and/or its affiliates.

# Creating Business Flags

Business flags may be created in a system for one of the following reasons:

- A message is received from an external system that initiates the creation of a business flag. In this case, logic in the external system has detected some situation that this product is being alerted about.
- Business logic in this product detects a situation that should be investigated or should act as a flag. In this scenario, there may not be any integration needed depending on the business rules.
- Business logic in this product detects a situation that another integrated product should be alerted about. In this scenario, the business flag record is used to send out information to the integrated product.
- A user manually creates a business flag based on knowledge of the affected entity. For example, a customer service representative may create a business flag as a result of contact with the customer.

**Creating a Business Flag from a Web Service**

The system supplies both a REST and SOAP inbound web service that may be used for an external system to initiate (or update) a business flag. The REST service is `F1-BusinessInsights`. The SOAP service is `F1-BusinessFlagSync`. In both cases, the web service invokes a "driver" service script whose ultimate responsibility is to determine the appropriate Business Flag Type based on the standard name or external standard name, and therefore the appropriate business object for the new business flag. Because different products may have different logic related to creating a business flag, the service script calls another service script linked to the maintenance object using the Business Flag Sync MO option.

The "driver" service script includes a "contextData" group, which in turn references the data area `F1-BusinessFlagContext`. This data area is to allow for an external system to provide additional information for a given business flag use case. For any integration that requires additional information to be passed in from the external system, a separate data area should be used to define the additional elements; and that data area should refer to `F1-BusinessFlagContext` as its extended data area.

The framework does not supply a Business Flag Sync service script, however individual edge applications supply a service script based on the use cases it supports out of the box.

> **Note**
>
> For products that are continuing to use XAI for external messages, the product also includes an XAI inbound service for the same Business Flag Sync Driver service script. Note that the product recommendation is to discontinue use of XAI and use inbound web services instead.

**Error Handling**

If there is a problem in trying to create a business flag based on incoming information, the Business Flag Sync Driver service script creates a special business flag record using the Business Flag Error Business Object. This is also configured on the maintenance object as an option. The framework product supplies the business object Business Flag Error (`F1-BusinessFlagError`) for this functionality. Refer to the business object description and configuration for more information.


Copyright © 2026, Oracle and/or its affiliates.

## Confidence

There may be use cases where a condition is suspected, but not confirmed. The originating system should be able to assign a "confidence" level to the business flag.

For example, Oracle Utilities Analytics Insights or other machine learning platforms will aggregate and analyze large amounts of data to identify potential insights that need investigation. These platforms can initiate a business flag with a **Suspected** confidence.

An application receiving this business flag may adjust the confidence to either **Confirmed** or **Rejected**.

- – Setting the confidence to **Confirmed** is the equivalent of a true positive.
  - – Setting the confidence to **Rejected** is the equivalent of a false positive.

Note that the application that receives a business flag is responsible for acting on the value based on business rules. Any updates to the confidence can be communicated to the other products that received the same business flag, if desired.

Additionally, creating a business flag in the application with a confidence of **Missed** provides a way to denote a missed insight. This is the equivalent of a false negative.

Because a utility implementation may have multiple applications installed that support business flags, the following guidelines are suggested for designing where the confidence flag should be updated.

- If Oracle Utilities Service Order Management is implemented, it has the responsibility of updating the confidence flag and communicating the update to other products.
- Otherwise, the assumption is that Oracle Utilities Customer Care and Billing owns field work orchestration and that it will have the responsibility for updating the confidence flag and communicating the update to other products.

No product logic is provided to enforce the above suggestions, however, the business objects supplied by the different edge applications will support the recommended implementation.

**Business Flag Updates from External System**

When the product that is responsible for updating the Confidence flag makes a change, it should initiate an outbound message to alert other products. On the receiving side, the same inbound web service and Business Flag Sync service script is responsible for the update. Refer to Creating Business Flags for more information.

# Setting Up Business Flag Configuration

The following topics highlight the general configuration steps required to use business flag functionality. Your specific product may supply additional functionality to support specific use cases for business flags. Refer to your specific product documentation and the library of business objects supplied for Business Flag in your product for more information.

- Standard Name Category Characteristic Type
- Standard Name Lookup
- Business Process Lookup
- Integration Configuration
- Defining Business Flag Types


Copyright © 2026, Oracle and/or its affiliates.

## Standard Name Category Characteristic Type

Define one or more categories for grouping your standard names into logical business groupings.

Navigate to the Characteristic Type page. Search for and select the Business Flag Category characteristic type (`F1-BUSFC`).

Define desired category values and descriptions to be used for the standard names.

## Standard Name Lookup

Navigate to the Extendable Lookup portal. Search for and select the **External Insights Standard Name** business object.

Define values that are recognized in the external systems that your implementation is receiving business flag details from.

Define a **Category** for the standard name that is appropriate for your product. Note that the category does not have to be in sync with standard name definitions in external products.

Select a **Class** for the standard name that is appropriate for its usage.

Refer to the embedded help for more information about configuring this object.

## Business Process Lookup

If your specific product supports configuring business processes that may be impacted by the existence of a business flag, they are defined as an extendable lookup.

Navigate to the Extendable Lookup portal. Search for and select the **Business Process** business object.

## Integration Configuration

The following points highlight configuration required to support receiving business flag information from an external source:

- Define a record for each External System that the product may be receiving business flag records from. This should be a value known by the external system and provided when new business flags are sent to this product.

When this product should initiate business flag information to be sent to an external system, configure one or more Outbound Message Type records. For each one, update the External System to configure how each outbound message type is communicated to the external system.

## Defining Business Flag Types

Refer to About Business Flags for an overview of business flag functionality.

To maintain the business flag types applicable to your product, open **Admin**, and then **Integration**, and then **Business Flag Type.**

This is a standard All-in-One portal.

The information captured on the business flag type depends on the business objects supported by your product. Refer to the embedded help text for more information.


Copyright © 2026, Oracle and/or its affiliates.

## Maintaining Business Flags

This section describes the functionality supported for viewing and maintaining business flags.

Refer to About Business Flags for an overview of business flag functionality.

Navigate using **Main**, and then **Integration**, and then **Business Flags**. You are brought to a query portal with options for searching for business flags.

Once a business flag record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Business Flag** zone provides basic information about a business flag. Refer to the embedded help for more information.

# Market Transaction Management

This section describes various entities and aspects involved in the configuration and maintenance of market transactions. The framework product supplies basic functionality to support the objects related to market transaction management. However more specific functionality (business objects and algorithms) for specific use cases are supplied by the edge applications.

- Understanding Market Transaction Management
- Configuring Market Transaction Management
- Maintaining Market Transactions

## Understanding Market Transaction Management

Several objects are provided to support market transaction management.

- Market Message Type. This object is used to define the different types of market messages, both inbound and outbound and to define the appropriate market message Business Object to use when creating objects of this type.
- Inbound Market Message. This object captures the instance of a market message received from an external source.
- Outbound Market Message. This object captures the instance of a market message that will initiate a message to an external source.
- Market Process Type. This object is used to define configuration related to processes that may be initiated to support steps that should be execute to support an inbound or an outbound market message. It defines the market process Business Object to use when creating market processes of this type. It may also define one or more market process event business objects to initiate events that are generated to support steps within the market process.
- Market Process. This object captures the instance of a market process that may be created to carry out business logic related to a specific inbound or an outbound market message.
- Market Process Event. This object is used to orchestrate individual events that may occur to support steps in a given market process.


Copyright © 2026, Oracle and/or its affiliates.

- Market Configuration. This object may be used to capture configuration needed to support market transactions. Each type of configuration should use a Business Object to define what information needs to be captured. Each type of configuration may define up to five key fields that may be used to uniquely identify an individual configuration instance.

# Configuring Market Transaction Management

The following sections describe market transaction management configuration options.

- Defining Market Configurations
- Defining Market Message Types
- Defining Market Process Types

## Defining Market Configurations

Each market configuration is used to define one or more options for a given type of configuration (based on a delivered business object).

The product defines a generic set of portals, described here, that may be used to define any type of market configuration. However, it may be that your implementation includes more specific configuration portals for certain types of market configurations.

Open this page using **Admin**, and then **Market Transaction Management**, and then **Market Configuration.**

You are brought to the **Market Configuration Query** where you need to search for the market configuration (i.e., its business object).

Once you have found the appropriate market configuration, select the value and you are brought to a standard All-in-One portal that lists the existing values for the market configuration. The standard actions for an All-in-One portal are available here.

## Defining Market Message Types

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Admin**, and then **Market Transaction Management**, and then **Market Message Type.**

This is a standard All-in-One portal.

The information captured on the market message type depends on the business objects supported by your product. Refer to the embedded help text for more information.

## Defining Market Process Types

This page is used to define market process types which control behavior of the resulting market processes and the valid market process events that could be triggered as a market process progresses through its lifecycle.

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Admin**, and then **Market Transaction Management**, and then **Market Process Type.**


Copyright © 2026, Oracle and/or its affiliates.

This is a standard All-in-One portal.

The information captured on the market process type depends on the business objects supported by your product. Refer to the embedded help text for more information.

# Maintaining Market Transactions

The following sections describe various pages available to view and work on market transactions.

- Maintaining Market Processes
- Maintaining Market Process Events
- Maintaining Market Messages

## Maintaining Market Processes

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Main**, and then **Market Transaction Management**, and then **Market Process.**You are brought to a query portal with options for searching for market processes.

Once a market process record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Market Process** zone provides basic information about a market process. Refer to the embedded help for more information.

#### Related Objects

The related object tab includes several zones that display various objects that may be related to the market process.

#### Related Market Messages and Processes

This zone displays any entries in the related object collection that are outbound market messages, inbound market messages or market processes.

#### Related Processes

This zone displays market processes that are linked to this market process using the explicit related process table. The market processes linked here are marked either as a parent, a child or a sibling market process.

#### Other Related Objects

This zone displays market process events linked to this market process. In addition, the zone displays any miscellaneous objects that are linked to the market process in the related objects collection, besides outbound market messages, inbound market messages and market processes.

## Maintaining Market Process Events

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

You can view this page when drilling into a market process event from the Market Process related objects tab.


Copyright © 2026, Oracle and/or its affiliates.

Once a market process record has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Market Process Event** zone provides basic information about a market process. Refer to the embedded help for more information.

#### Related Objects

The related object tab includes a zone that displays all entries in the market process event's related object collection.

## Maintaining Market Messages

Refer to Understanding Market Transaction Management for an overview of market messaging functionality.

Open this page using **Main**, and then **Market Transaction Management**, and then **Market Message.**You are brought to a query portal with options for searching for market messages. Both inbound market messages and outbound market messages may be queried here.

Once a market message record has been selected, you are brought to the appropriate maintenance portal (inbound market message or outbound market message) to view and maintain the selected record. The inbound market message portal and the outbound market message portal have similar portal tabs and zones.

The **Market Message** zone provides basic information about the market message. Refer to the embedded help for more information.

#### Related Objects

The related object tab includes several zones that display various objects that may be related to the market process.

#### Related Market Messages and Processes

This zone displays any entries in the related object collection that are outbound market messages, inbound market messages or market processes.

#### Other Related Objects

This zone displays any miscellaneous objects that are linked to the inbound / outbound market message in the related objects collection, besides outbound market messages, inbound market messages and market processes.

# Content Migration Assistant (CMA)

This chapter describes the Content Migration Assistant (CMA), a facility to enable the export of data from one environment to another.

> ⚠ **Caution**
>
> This chapter is intended for users responsible for creating environments for testing.

- Understanding CMA
- CMA Configuration


Copyright © 2026, Oracle and/or its affiliates.

- The CMA Execution Process
- Running Batch Jobs
- CMA Reference

# Understanding CMA

The Content Migration Assistant (CMA) provides implementers with a flexible, extensible facility for migrating data from one environment to another (e.g., from a development environment to a production environment). Data is exported from the source system to a file. The file can then be checked in to a version control system for reuse, or can be immediately imported into the target system and applied.

> **Note**
>
> As used in this chapter, *source* systems are those on which export-related activities are conducted and *target* systems are those on which migration updates are to occur.

The following points highlight some common use cases for using this tool:

- Seeding a newly created environment with configuration data. For example, if you are creating test environments, you can import configuration data from a development environment. In this case, you will probably set up a "copy all configuration" type of migration. Another use case is if you want to troubleshoot an issue in production, you may want to copy all production configuration to a test environment.
- Updating an existing test environment with new configuration or updates to configuration from a development environment. In this case your set of objects will be more targeted and you are doing more of an incremental migration.
- Promoting tested configuration to your production environment.
- Migrating selected master and transaction entities from one environment to another for testing purposes.

Here are key aspects of the import process that apply to all cases:

- When records exist in both the target and the source, the CMA tool will fully **replace** the object with the version from the source. Special consideration should therefore be given to records whose primary key is system generated to ensure the correct entity is being replaced.
- While typically the source version of an object is applied directly onto the target environment, CMA does provide a way to Adjust Imported Data.
- The import process **does not handle deletion** of entire entities in the target environment. Refer to Configuration Data Deletion for more information.

Configuration steps are mainly used to define the data to migrate at the source system but also involve tasks to be performed at the target environment. Note that the products provide base delivered configuration that may be used as is or used as a template for building more specific configuration for a given implementation. At a high level, configuration tasks involve the following entities:

- System wide settings are defined in the **Migration Assistant Configuration** master configuration record.


Copyright © 2026, Oracle and/or its affiliates.

- Each type of record that may be copied requires a **Migration Plan**. The migration plan is used to identify the maintenance object (MO) for the record (using a business object) and allows for instructions to specify related records that may be included in the migration.
- A **Migration Request** is used to define the data to include in a given migration export.

Refer to CMA Configuration for more information.

Once you have a migration request defined to include desired data to export, the next steps are to export from the current environment and import to the target.

- The **Export** process includes all the steps needed to select records to be exported from the source environment and create the export file. For more information, refer to Exporting a Migration.
- The file created by the export, which is a BINARY file, needs to be transferred from the export directory to the import directory. The transfer needs to be done in such a way as to preserve the file structure. Refer to Additional Migration Considerations for more information.
- The **Import** processes include all the steps needed to read an imported file, compare the data in the file to the data in the target, review the proposed changes and apply the updates. For more information, refer to Importing and Applying a Migration.

The topics in this section provide a general understanding of the Content Migration Assistant (CMA) tool.

- Types of Migrations
- Migrating Configuration Data
- Migrating Business Data
- Data with System Generated Primary Keys
- Importing Data that References a User
- Importing Environment-Specific Data
- Configuration Data Deletion
- Additional Considerations
- Versions and CMA

# Types of Migrations

The Content Migration Assistant is used for two general types of migrations: wholesale and targeted. The following topics provide some additional information about these concepts.

### Wholesale Migrations

Wholesale migrations are used when migrating all the configuration and/or administrative data from one environment to another. For example, a wholesale migration might be used when migrating administrative data from a development or test environment to a production environment.

A wholesale migration may be comprised of one or more migration requests that in total include all the administrative data to move. Migration plans used in wholesale migrations may be designed to omit subordinate instructions related to explicit foreign keys that are identified through constraints as they are not needed, assuming that the data they are referring to will also be included in the migration.


Copyright © 2026, Oracle and/or its affiliates.

With the ability to group migration requests, the expectation is that implementations follow these guidelines:

- Multiple migration requests using the Criteria-based or Entity List migration request classes are used to group information logically together to allow for more reuse.
- A Group migration request is used for the export. This allows for one data set to export and one data set on the import side, simplifying the process. Note that depending on the amount of data, this may be a large import set to process. An implementation may find it easier to create multiple migration requests that break down the process into several steps.

You should consider that the framework product provides base migration requests and your specific edge product may provide base migration requests as well that may or may not include framework migration plans. Using the product provided migration requests is beneficial with respect to maintenance. As features are added to the product (including new administrative maintenance objects), any impact on CMA should be included in the product owned migration requests. If your implementation introduces new custom administrative maintenance objects that should be included in CMA, then custom migration plans and a custom migration request should be added. Your implementation can build a Group migration request that includes the base migration request and your custom migration requests to have a consolidated export.

> **Note**
>
> Refer to Framework Provided Migration Objects for information about migration requests provided in the framework product. Refer to your specific product's documentation for information about addition base provided migration requests.

#### Targeted Migrations

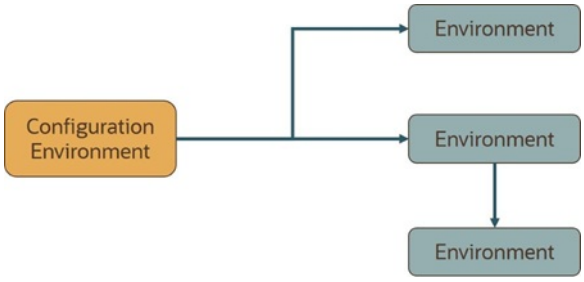
A targeted migration refers to migrating a specific subset of data from one environment to another. Migration plans used in targeted migrations are designed to be self-sufficient and include all necessary subordinate instructions needed to ensure that the exported entity can be imported without encountering referential integrity errors. With this type of migrations, any relationship explicitly excluded from the migration plan assumes the related entity already exists in the target environment. Examples of targeted migrations include:

- Migration of a new portal, its zones, and its application service.
- Migration of all outbound message types.
- Migration of selected accounts and their related master and transactional entities to a lower environment for testing purposes.

## Migrating Configuration Data

A main use of CMA is to migrate configuration data, where the source environment is considered a kind of system of record and the target environment is getting updates based on changes to the source environment. Alternatively, you may be seeding a new environment. The expectation is that the target record is updated to look like the source record.


Copyright © 2026, Oracle and/or its affiliates.

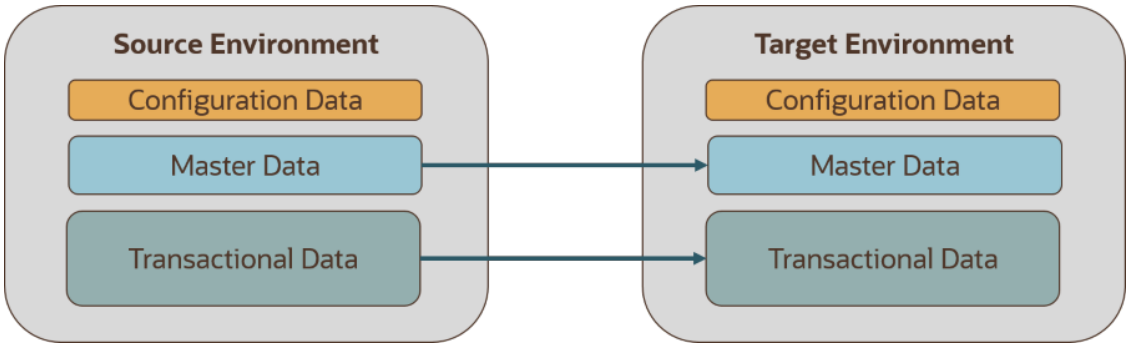



Configuration data may include a mixture of records that have an owner flag and ones that do not. The comparison step of the import process will generate appropriate insert or update SQL statements for the following data found in the export:

- Configuration data in a maintenance object with no owner flag. This is purely implementation data.
- Configuration data in a maintenance object with owner flag, where the owner is **Customer Modification**. For example, implementer-specific business objects.
- Configuration data in a maintenance object with owner flag, where the main record is owned by the product but where a child record exists with an owner of **Customer Modification**. For example, implementer-specific algorithms added to a base owned business object.
- Customizable fields in a record that is owned by the product. For example, the priority of a based owned To Do type.

## Migrating Business Data

CMA may be used to perform a targeted migration of selected master entities and their related transactional data from one environment to another. For example, migrating a subset of accounts and their related data for testing purposes.




The following points highlight key differences between business data and configuration data that call for special considerations when designing your business data migrations:

- Migration of business data typically involves a much higher number of records in a data set compared with a configuration-only data set. While there is no explicit limit to the size of a data set, some reasonable limit has to be assumed.
- Having large sets of entities depend on relatively few entities in the same data set significantly slows the import process. This situation occurs when mixing configuration data, master data and high volume transaction data in a single migration. We therefore recommend designing separate migrations for these data classes. Refer to the **Reduce Dependency Between Migration Objects** section below for more information.


Copyright © 2026, Oracle and/or its affiliates.

- All business entities have a system generated key. Refer to Data with System Generated Primary Keys for more information.

The following sections expand on topics and concepts related to business data migration.

### Reduce Dependency Between Migration Objects

CMA leverages the application's referential integrity and business rules to ensure entities remain compliant and valid when imported into another environment. These rules are there to prevent an entity from being imported when any of the entities it references do not exist in the target environment. When both the referencing entity and the entity it references are part of the same data set they establish a **dependency** that dictates the sequence in which these entities should be processed in order to be successfully imported. Note that the tool supports even the more complex and rare situations wherein the dependency between entities is cyclical, i.e. entity A (directly or indirectly) refers to entity B while entity B (directly or indirectly) refers back to entity A, for which sequential processing is not enough to guarantee a successful import.

The tool uses foreign key relationships between entities in the same data set to identify sets of dependent entities and group them into separate migration transaction records. Note that these foreign keys are defined in the application's data model repository and not at the database level for greater flexibility. The larger the migration transaction the more complex the process is to successfully import all its objects. So while the tool is designed to support all types of dependencies (sequential and cyclical), handling very large sets of dependent objects may incur a significant performance cost.

While configuration entities tend to be highly inter-dependent, the total number of entities that comprise the entire set of configuration objects is relatively small. Therefore, the impact on performance such dependencies have when importing a configuration-only data set is negligible. On the other hand, configuration entities tend to be referenced on many master and transaction entities as they control many aspects of the latter's business rules. As such, mixing configuration and business data in the same data set is likely to form extremely large sets of business entities that dependent on a small number of configuration entities. It is therefore strongly recommended to import or set up configuration data in the target environment prior to importing business data into it.

In a similar way, some transactional data tends to be of a much higher volume compared to the master data entities they reference. Here too, it is strongly recommend to migrate master data entities prior to their high volume related transactional data. For example, many interval measurement records (transaction data) reference the same measuring component (master data). Migrating a large volume of measurement data along with their measurement component would result in a very large migration transaction that may take long to import. Importing the measurement components prior to migrating their interval measurements will eliminate unnecessary dependencies within the measurements only data set and significantly improve the performance of the overall process.

### Reasonable Data Volume

The overall volume of all business entities to migrate in a single data set should be reasonably sized. For example, importing several hundreds of accounts and their related master and transactional data is considered a reasonable size. Migrating too much data may reach physical and performance limitations of the tool.

You may use any supported migration request methods for describing the entities to export. If you opt for an Entity List migration request, consider using the Collect Entity zone to build the list of entities as you browse them on their respective portals.


Copyright © 2026, Oracle and/or its affiliates.

### Single Source of Data

CMA uses an entity's prime key to determine if it is new to the target environment and therefore should be added or refers to an existing record and therefore needs to be replaced with the new version. All business entities have system generated keys, which are environment specific. It is therefore possible to have different entities in separate environments having the same prime key. When migrating entities whose keys are system generated, it is strongly recommended to migrate data to a target environment from a single source environment and avoid creating such entities in the target environment using the application. This practice ensures that prime keys for an imported entity are always in sync with their source environment. Refer to Data with System Generated Primary Keys for more information and considerations.

### No Deletion

CMA does not handle deletion of entities of any kind, whether they are configuration or business entities. If a test needs to be repeated using an initial snapshot of the data, you should restore your target environment to a backup taken before test data was imported and import the latest version of the test data. If there is no need to delete previously imported entities, you may keep reloading test data from a single source environment as needed.

### Bulk Import Mode

By default, the import process creates a migration object for each imported entity. This allows for granular reporting and error handling at the entity level. When importing a high volume data set of business entities, this granular management has a performance toll. In this situation, you may want to use the **Bulk Import** option on the migration data set import request. In this mode, a single migration object record is created for multiple entities of the same maintenance object. In the same way, a single migration transaction record is created for multiple logical transaction groups. Using this option reduces the migration object management effort throughout the process and thus results in better performance.

Note that each entity is still individually compared and validated, as in regular import processing, but if one entity is invalid the entire migration object is not applied hence impacting all entities grouped together with it. Bulk import mode is useful when importing a large set of data from a validated data source such that almost no errors are anticipated.

This option is only supported for master and transaction maintenance objects, i.e. it is not applicable to configuration migrations.

### Insert Only Mode

By default, the comparison step of the import process needs to determine whether the imported entity is new and therefore should be added, represents a change to an existing entity and should be updated or the entity is unchanged. When importing a high volume data set of business entities that are all new to the current environment, this check is time consuming and can be avoided. In this situation, you may use the **Insert Only** option on the migration data set import request to indicate that all imported entities are assumed to be new additions to the current environment. By doing so, the import process avoids unnecessary steps to determine whether the entity should be added or updated and thus contributes to a faster import process.

This option is only supported for master and transaction maintenance objects, i.e. it is not applicable to configuration migrations.


Copyright © 2026, Oracle and/or its affiliates.

### Separate Batch Processes for Managing Business Migration Objects

By default, the same import related batch processes manage configuration and business data migrations. Typically, business data migrations involve high volume of records compared to much lighter configuration data sets. Processing them together by the same batch process may slow down the performance of configuration migrations, preventing them from completing faster and more frequently. The issue mainly affects test-like environments where mixed data class migrations are more common: configuration data is imported from a lower environment and large test data is imported from a higher environment. You may adjust the base product's configuration in such environments to benefit from separating the import processes for configuration and business data.

Note that the ability to separate import processes applies to migration objects only because of their volume. Migration data set and transaction records are of low volume and as such remain managed by the same batch processes.

Designated batch processes are provided for importing migration objects containing business data but they are not used by default:

- F1-MGOPB - Migration Object Monitor (Business)
- F1-MGOAB - Migration Object Monitor (Business) - Apply

Follow these steps to use these separate designated batch processes for business data:

- Update the **Migration Object Business Data** (F1-MigrObjectBus) business object to reference the business data related batch controls on the following statuses:
  - **Pending, Error Applying, Needs Review** - Migration Object Monitor (Business) - F1-MGOPB
  - **Approved** - Migration Object Monitor (Business) - Apply - F1-MGOAB
- If your organization uses event driven job submissions for CMA batch processes then refer to Running Batch Jobs for additional configuration steps.

## Data with System Generated Primary Keys

For records with system generated keys, there is an issue if the target environment already has a record with the same key but doesn't represent the same record. The tool shouldn't be just updating that target record with the view of the source in case this will break foreign key relationships that exist in the target.

The tool provides support for administrative data with system-generated primary keys. The logic relies on the maintenance object to use a method that looks at other attributes of the record (considered a "logical key") to detect whether the record being migrated already exists in the target region or not. The examples in this section are based on the Attachment maintenance object. Common attachments are considered administrative data. The attachment MO uses the file name and the creation date as the "logical key".

Imagine a common attachment for the "standard rate codes" file exists in a source region with the key 123456789. The table below highlights possible situations at the target region and actions supported in CMA.

| Scenario | Target Situation | Action | Comments |
|---|---|---|---|
| 1 | No matching record | Record can be added with key 123456789. | |


Copyright © 2026, Oracle and/or its affiliates.

| Scenario | Target Situation | Action | Comments |
|---|---|---|---|
| 2 | Record exists with key 123456789 and logic confirms that it is also the "standard rate codes" attachment. | Record can be updated. | |
| 3 | Record exists with key 123456789, but logic detects that it is not the "standard rate codes" attachment. | Record is not updated. An error is issued. | The system cannot update this record because it's not the right attachment record. |
| 4 | The system detects that another attachment record exists for the "standard rate codes" attachment with a different ID. | Record is not updated. An error is issued. | Assumption is that the record was created directly in the target or was copied from a different source. |

The use cases described in scenarios 3 and 4 above would require key mapping to keep track of the id from the source to the id in the target so that any other records from the source that reference this key as a foreign key would be updated as part of the migration. This functionality is not supported.

Scenarios 1 and 2 above are supported for maintenance objects that use the method to detect the logical key.

> **Note**
>
> If a maintenance object with a system generated key does not supply a method to detect the logical key, CMA will update an existing record with the same ID. For maintenance objects in the framework that provide this method, refer to Framework Provided Migration Configuration. For your specific edge application, refer to the CMA addendum for information about support for data with system generated keys.

The product recommends that an implementation establishes a migration strategy such that records with system generated keys are always created in the same region and always follow a standard migration path for promoting the data from this source region to other regions. Following this strategy, you would minimize or eliminate the possibility that a record for the same logical key is created in multiple places such that different IDs would be generated as described by scenario 4 above.

**MOs with a Mixture of Administration and Non-Administration Data**

There are some MOs that contain a mixture of master or transaction data and administrative data. The Attachment is an example of this. The product supports common attachments and owned attachments. Owned attachments are records that are specific to its owner. The owner could be master or transaction data and its attachments are therefore considered master or transaction data. Owned attachments are not candidates for migration using CMA. Common attachments on the other hand are considered administrative data and may be candidates for migration using CMA. For these use cases, an implementation may follow the suggested strategy of only creating the administrative data in one region so that IDs for common attachments are not reused. However, it is reasonable and expected that owned attachments are being created in the target region and may receive a system generated key that matches the key of a common attachment from the source region.

To try to minimize this issue, the system includes special logic to be used by any MO that may contain administrative data mixed in with master or transaction data. This special logic


Copyright © 2026, Oracle and/or its affiliates.

generates the key of an administrative record with a zero (0) in the middle of the key and ensures that the keys for master and transaction data do not include a zero in this spot. For maintenance objects in the framework that use this method, refer to Framework Provided Migration Configuration. For your specific edge application, refer to the CMA addendum for information about additional maintenance objects that may be in this category.

## Importing Data that References a User

Although the tool supports copying users from one environment to another, the assumption is that it will be more common for implementations to define their valid users for each environment separately. In fact, the default CMA configuration for copying security configuration assumes that users are not being copied.

There may be times were a record is being copied from one environment to another where a user is referenced on the record but that user is not valid in the target environment. For example, if a batch control is configured as Timed with a user referenced, if this batch control is copied to another environment, what should happen if that user is not found? You may not want that user defined in the target environment. To handle this, the product provides support for a Default User that you can define in the migration assistant configuration. If this default user is configured then any time the CMA import process recognizes that a record references a User and the user does not exist in the target environment, it updates the record to instead refer to the default user.

> **Note**
>
> Please note that if you are trying to copy entities that includes valid users, you should ensure that all the users are copied to the target environment first. Otherwise, the entity will be updated to refer to the default user, which may not be your intention.

> **Note**
>
> If you copy records that reference a User within an XML field or CLOB, the CMA input tool is not able to detect that. A pre-compare algorithm is recommended if this use case exists in your implementation.

## Importing Environment-Specific Data

Configuration entities that include references to environment-specific values should be imported with extreme care. Unintentionally overriding these values with data sourced from another environment may disrupt the proper function of the application. Note that these records may be initially imported from another environment but are typically adjusted manually to reflect current environment settings and should not be updated again by an import.

The sole purpose of some maintenance objects, like Message Sender, is to define environment-specific settings. As such, they are easily identifiable and included in a designated integration related migration request (F1-IntegrationConfig) that can be used carefully and as needed.

However, general purpose maintenance objects, like Extendable Lookup and Master Configuration, are designed to define various types of configurations where some may contain environment-specific information. For example, while most extendable lookup records do not contain environment-specific settings, the File Storage Configuration (F1-FileStorage) extendable lookup includes object storage definitions that are environment dependent.


Copyright © 2026, Oracle and/or its affiliates.

The following mechanism may be used to prevent an unintentional update of such records by a CMA import. It identifies the records containing environment-specific data and allows for them to be added but not updated as follows:

- Use the **Environment Specific** option to mark business objects that define environment specific information. For example, the business object for the File Storage Configuration extendable lookup is marked with this option.

> **Note**
>
> This option is available for Extendable Lookup and Master Configuration business objects but may be configured to be valid for other maintenance objects as needed.

- The product supports a generic migration plan **pre-compare algorithm** that prevents an update of an entity that its business object is marked as containing environment-specific data using the designated option.

> **Note**
>
> The algorithm is added to the Extendable Lookup and Master Configuration default migration plans and can be added to other migration plans as needed.

In addition, the following changes were made to keep such entities in environment-specific related migration requests:

- The wholesale **General System Options** migration request excludes extendable lookup and master configuration records that their business object indicates they contain environment-specific information.
- The wholesale **Framework Integration Configuration** migration request include only those extendable lookup and master configuration records that their business object indicates they contain environment-specific information.

## Configuration Data Deletion

The import process does not handle deletion of entire entities in the target environment. When importing data from a source environment to a target environment, the import process for the migrated records is able to identify objects to add and objects to change. There is no mechanism for comparing the target environment to detect entities that were previously imported and no longer exist in the source system. In the same way, there is also no mechanism for indicating that records in the target environment should be deleted. The absence of those records in the import is not enough because the migration may be only importing a subset to add or update.

Note that CMA does orchestrate the deletion of child rows of an object as a result of a comparison. This is only applicable to child records that are owned by the implementation.

If data on the target system must be deleted, users must delete the records in the target accordingly.

This section describes available tools to assist with this task.


Copyright © 2026, Oracle and/or its affiliates.

### Delete Entities Online

You may review and delete previously imported configuration entities using the Configuration Deletion portal.

### Delete Entities In Batch

You may submit a request to delete configuration entities in batch. The process involves uploading a list of configuration entities from a CSV file onto a request record that is processed in batch by the request monitor process.

To do so create a request record using the **Delete Configuration Entities** (F1-DeleteConfigEntitiesType) request type. This option allows you to upload a list of entities to delete from a CSV file onto the request record and submit the request for batch processing when ready. When processed in batch, the request deletes all the entities in the list as a single transaction.

The file with the entities to delete should be prepared with the following header columns: maintenance object, key1, key2, key3, key4, key5. Add rows after the header row for each configuration entity to delete.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Maintenar | key1 | key2 | key3 | key4 | key5 |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

> **Note**
>
> The file should be of a reasonable size to allow the list to properly upload online. If the upload step times out use smaller files.

The deletion process abides to the application's standard referential integrity rules and therefore the order in the list should reflect data model relationship dependency.

The process stops at the first error, i.e. it does not accumulate exceptions. The user needs to review the exception, correct the issue, and resubmit the request as needed.

# Additional Considerations

The following sections describe some additional considerations for CMA.

### File Transfer Considerations

When moving the export file between systems, use the binary transfer option of whatever tool you use to move the file so that line-end characters are not converted from Linux-style to Windows-style or vice versa.

It is recommended to avoid using 'txt' for the export file's extension (defined in the master configuration). That file extension by default implies a non-binary file and tools that perform file transfer may treat this as a non-binary file unless explicitly stated. The recommendation is to define 'cma' as the extension. This is not a recognized file extension and most file transfer tools will transfer the file as is.


Copyright © 2026, Oracle and/or its affiliates.

Note that if the file gets converted, there are two likely outcomes - either a numeric conversion error, or a buffer under-run error may be received when attempting to import the file.

### Multi-Language Environment Considerations

If your implementation uses a language other than English, it means that migrated administration objects may have multiple language rows (because English is always enabled). There are some important points with respect to multiple languages and CMA:

- As described in User Language, there are steps to follow when supporting an additional language. The steps outlined in that topic highlight that for system data, translation of the strings may be provided via a language pack provided by the product or may be the responsibility of your implementation. In either case, this effort is non-trivial and will have its own established plan. The expectation is that the translation of the system data is applied for each region at your implementation site. CMA should not be used to create a new language in a target region.
- For administrative / control data that your implementation develops as part of your project, the expectation is that descriptions for your supported language are entered in the region that is considered the source region used to promote changes to regions in the "chain". For example, control data is entered in a development region and promoted to a test with the supported language enabled in both regions.
- What if you export data from a region with more languages enabled than your target? This scenario is perhaps a case where the source region is a type of test or playpen region where the additional language is enabled for other purposes. In this case, if the language code doesn't exist at all in the target region, the import will produce an error given that the code is invalid. If the language code exists but is not enabled, this will cause the extra language rows to be inserted in the target region, but will not cause any issues. They are simply ignored.
- What if you export data from a region with fewer languages enabled than your target? In this situation, the import process will only create language rows for the languages that were copied from the source. It will not automatically create language rows in the target as part of the import. For this situation, the recommendation is to run the **New Language** batch program (`F1-LANG`) that creates any missing language entries.

## Versions and CMA

This topic addresses two separate version related topics with respect to using CMA between two environments.

- Different versions of the application
- Different CMA file format versions

### Different Versions of the Product

CMA was developed with the main requirement of copying data between two environments that are on the same version of the product. For example, copying configuration data from a development environment to a test environment on the same product version. It is highly recommended that you limit your use of CMA to this scenario. If it is impossible for you to ensure that both environments have the same product version, then this section highlights the possible challenges you may face when trying bring data in from a different product version.

In general, there isn't anything about how CMA functions that would limit your ability to copy data between two different versions of the product. However, there are several ways that your product may have enhanced or changed functionality for certain objects such that the data captured in a different version may not be compatible with the changes.


Copyright © 2026, Oracle and/or its affiliates.

Our products follow strict standards for upgrading clients to ensure backward compatibility. Upgrade scripts in the blueprint process know what was changed from one release to another and update the data accordingly, where applicable. When using CMA to move data from one version of the product to another, you are bypassing these upgrade scripts. It means your data is not being adjusted as expected. It's not recommended unless you are certain that the data of an object from one version is compatible in the other version.

The following points highlight this in more detail.

- For objects whose structure and rules have not changed between the two versions, there should be no issues.
- For objects that have changed between the two versions, there are scenarios that may cause additional steps.
  - If the target environment is a later version and includes a new optional column on an object that was not present in the source, the import should go through without any issues.
  - If the target environment is a later version and includes a new required column on an object that was not present in the source, a 'pre-compare' algorithm could be used to populate the new required value. Your implementation can provide this algorithm to populate the data according to your business rules.
  - If the target environment is an earlier version than the source, we cannot guarantee that there won't be specific issues encountered with specific objects.

> ⚠ **Caution**
>
> When importing data from a different environment and bypassing upgrade scripts, it may be that nothing about this product change causes an error in CMA. But because the data has not been updated as expected, it's possible that something will not work properly in the application that is not easy to detect.

Another issue you may encounter is that if system owned data is part of your import file, it's possible that the product has made changes to that data as part of an upgrade and your version of the data is different. Because CMA tries to make the target data look like the source data, when it sees a difference in product owned data, it won't proceed with importing that data. Some examples:

- Your import has included all portals, including system owned portals. In the target version, one of the zones on the portal is no longer supported by the product and is no longer linked. In your import, the zone is still linked and the system attempts to add that zone back to the portal. Because the data is system owned, it marks it as "Cannot Apply". In this case, you don't need to do anything. You can just skip this record.
- If in the previous use case you had added a custom zone to the portal and are trying to copy in that configuration, you will not be able to do it because of the system data differences. In this case you will need to manually add the custom zone to the portal in the target environment.

### File Format Version

If your implementation has decided that you are going to use CMA across product versions, given all the cautions described in the previous section, this section provides information on the rare case where CMA has a change in how it writes the export file.

When creating the export file, the format of the file is specific and the CMA import step knows how to read the file. It is rare, but there are times when the product makes an enhancement to


Copyright © 2026, Oracle and/or its affiliates.

CMA that requires a change in the structure of the export file at which point the 'version' of the CMA file is incremented.

The CMA file format version has no impact to migrations that are exported and then imported into the same version of Oracle Utilities Application Framework. However, if your implementation has different environments on different versions of the product where the CMA file format is also different, then the system has functionality to cater for these differences.

- If you are exporting data from a lower file format version to a higher file format version, there is nothing special that you need to do. The export step captures the file format version and the import step reads this information and processes the file accordingly.
- If you are exporting data from a higher file format version and the product currently supports exporting to a lower file format version, at export time you are prompted for the earlier file format version. You may indicate the you want the earlier file format version when creating the migration data export.

> **Note**
>
> The product supports exporting to an earlier file format version for a reasonable amount of time, but not in perpetuity. If you don't see any option in the user interface to export to an earlier file format, it means that it is not supported in that release.

# CMA Configuration

The following sections describe tasks required for CMA configuration.

- Master Configuration - Migration Assistant
- Migration Plans
- Defining a Migration Request
- Identifying Tables to Exclude From Migrations
- Configuring Custom Objects for Migration

## Master Configuration - Migration Assistant

The **Migration Assistant Configuration** master configuration record defines system wide configuration used in both the source environment and the target environment.

As described in Importing Data that References a User, if you have records being imported that may reference a user does not exist (and shouldn't exist) in the target environment, define a Default User here.

In both the source environment and the target environment, the system needs to know the location of the export directory and the import directory along with the expected file suffix. For these fields, implementations may also rely on the system defaults. If no Migration Assistant Configuration record is found, the system assumes that there is an entry defined in the system's substitution variable list for `F1_CMA_FILES`. Further it defaults the values as follows:

- Export directory is the value for this variable plus `"\export"`.
- Import directory is the value of this variable plus `"\import"`.


Copyright © 2026, Oracle and/or its affiliates.

- File suffix is set to **cma**

Refer to Referencing URIs for more information about the substitution variable list.

For additional information about specific fields in the master configuration, refer to the embedded help.

> **Note**
>
> This record can be updated at any time to change details. The new configuration takes effect on all subsequent exports and imports.

# Migration Plans

A migration plan defines one or more types of objects that are eligible for migration. It is essentially a set of instructions describing how the data to be exported is structured, allowing objects to be migrated together as a logical unit to ensure consistency and completeness.

The migration plan is always related to a given maintenance object, defined using its primary instruction. Multiple migration plans may exist for a given maintenance object if there are different requirements for migrating records in that MO under different circumstances. For each migration plan, you decide if additional data should be included as part of an object using subordinate instructions. For example, when copying a single algorithm type, should the system only copy that algorithm type record, or should it also copy all its algorithms and its related script (if applicable)? Refer to Types of Migrations for more information.

The following topics describe defining a migration plan as well as other topics for a migration plan.

- Defining a Migration Plan
- Understanding the BO Filtering Process
- Migration Plans for Objects with XML-Embedded Links

## Defining a Migration Plan

To view or define a migration plan, navigate using **Admin**, and then **Implementation Tools**, and then **Migration Plan.**

Use the **Migration Plan Query** portal to search for an existing migration plan. Once a migration plan is selected, you are brought to the maintenance portal to view and maintain the selected record.

> **Caution**
>
> Important! If you introduce a new migration plan, carefully consider its naming convention. Refer to System Data Naming Convention for more information.

The following points provide information about defining **Instructions** for a migration plan.

The **Instruction Sequence** uniquely identifies the instruction. The recommendation is to use increments of 10 to allow insertion of other instructions in the future.


Copyright © 2026, Oracle and/or its affiliates.

Select **Primary** for the first **Instruction Type**. All migration plans must contain one and only one primary instruction. All subsequent instructions require a **Subordinate** instruction type. In this case, the **Parent Instruction Sequence** must be entered. This number, used to maintain the defined relationships in the exported data, must match an instruction sequence number at a higher level in the hierarchy.

The instruction **Description** provides a business description of the instruction.

Select a **Business Object** (BO) to define the type of object from which data will be derived.

> **Note**
>
> Though BOs are specified in each instruction, it's important to understand that each BO is used only for filtering purposes. The migrated data set comprises the complete contents of the *maintenance object* that the business object structure is defined against. For a more detailed explanation of this, see Understanding the BO Filtering Process.

> **Note**
>
> Refer to Identifying Tables to Exclude From Migrations for information about defining child tables to always exclude from a migration.

**Traversal Criteria** is used to define the relationship between each of the objects in a migration plan. The system provides three options to define how the child object is connected to the parent object so the system knows how to traverse from one object to another. **Traversal Criteria Type** options are **Constraint**, **SQL** and **XPath**. The following points explain each option:

- **Constraint** allows you to select a table constraint that represents a given record's relationship to another record in the system via a foreign key constraint defined in the meta-data. If **Constraint** is selected, the following additional fields are enabled:
  - **Constraint ID** is a unique identifier for the constraint. The search will show the valid table constraints for the MO of the instruction's BO and the MO of the parent instruction's BO.
  - **Constraint Owner** is used to define the owner of the constraint. This is populated automatically when selecting a constraint from the search.
- **SQL** lets you specify SQL join criteria between the parent instruction's object and the child object in the **SQL Traversal Criteria**. The syntax of the the traversal criteria is a WHERE clause (without including the word WHERE). When referring to a field on the parent instruction's object, use the syntax `#PARENT.TABLE_NAME.FIELD_NAME`. When referring to a field on the current instruction's object, use the syntax `#THIS.TABLE_NAME.FIELD_NAME`. For example, the following statement is used on a migration plan for Business Object, where the parent instruction is the BO and the subordinate instruction is used to reference the UI Map that is referred to as a BO option with the option type `"F1DU":#PARENT.F1_BUS_OBJ_OPT.BUS_OBJ_OPT_FLG = 'F1DU' AND @trim(#THIS.F1_MAP.MAP_CD) = @trim(#PARENT.F1_BUS_OBJ_OPT.BUS_OBJ_OPT_VAL)`.
- The **XPath** option lets you apply syntax in an XPath expression referencing elements in the instructions' referenced business objects. This is entered in the **XPath Traversal Criteria**. For example, the display map collection statement in the SQL example noted above would be written as follows in XPath: `#this/mapCd = #parent/businessObjectOption/businessObjectOptionValue AND #parent/`


Copyright © 2026, Oracle and/or its affiliates.

`businessObjectOption/businessObjectOptionType = 'F1DU'`. This technique allows foreign key references that are mapped inside an XML column to be referenced.

> **Note**
>
> The #parent expressions may access elements that are stored in an XML column and described using mapXML and mdField. However, the #this expressions must refer to fields available in the business object using the mapField reference.

Defining **Next Migration Plan** provides the ability to indicate that in addition to copying the object defined in the instruction, any additional instructions included in that referenced migration plan will also be included in an export.

The **Algorithms** grid contains algorithms associated with each instruction. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the Sequence in which they should execute.

| System Event | Optional / Required | Description |
|---|---|---|
| `Pre-Compare` | Optional | Algorithms of this type may be used to adjust the data after it is moved to the target system. These may only be defined on the primary instruction.<br>Refer to Adjusting Imported Data for more information. |
| **Import** | Optional | Algorithms of this type are no longer supported. |

## Understanding the BO Filtering Process

Migration plan instructions require the definition of a business object to provide CMA with information about the record related to the instruction.

If the business object is the physical business object for the maintenance object, then CMA assumes that the instruction applies to all records that satisfy the traversal criteria. CMA recognizes the physical BO by comparing the BO to the value defined in the maintenance object option. If the business object defined is not the physical BO, then CMA will limit the records in the instruction to those that explicitly reference this BO or reference a child of this BO as its identifying BO value. (In other words, this BO must be in the parentage hierarchy of the records to be included in the instruction.)

> **Note**
>
> Unlike Bundling, CMA does not use the BO schema to drive what data is copied for a given record. The BO is only used as a filtering mechanism for selecting records. Refer to Identifying Tables to Exclude from Migration for information about how to ensure a child table is not included in a migration.


Copyright © 2026, Oracle and/or its affiliates.

For example, if you define a migration plan for Master Configuration and use the physical business object for the instruction (`F1-MstCfgPhysicalBO`) then all master configuration records are considered for the instruction. If instead the business object you define is Migration Assistant Configuration (`F1-MigrationAssistantConfig`) then only the record related to this business object is included in the instruction.

## Migration Plans for Objects with XML-Embedded Links

When migrating objects where foreign key references are captured in the object's XML based field, subordinate instructions are needed to define the foreign key references in order for CMA to understand the relationships. This is in contrast to direct foreign keys where CMA can determine the relationships using constraints. The instructions provide two purposes. Defining instructions ensures that the related objects are included in the migration, if appropriate. If you know that the related objects are included because of other criteria, the instructions allow CMA to group related objects into transactions. This helps in the apply process at import time to ensure that related objects are grouped together. However, the apply process includes iterative steps to try to overcome dependencies like this so defining the instructions is not critical for this purpose.

The following are options for creating migration plans with XML-embedded links:

- One option is to use the specific logical (business) BO in the primary instruction to define the object you are copying. With this option, the subordinate instructions may use XPath criteria to define the related foreign key. When this approach is used, a separate Migration Plan must be created for each logical BO. (Refer to Understanding the BO Filtering Process for more information.) This option would only be used in isolated cases.
- Another option is to create a migration plan that uses the Physical BO as the primary instruction, and then include a subordinate instruction for the real logical BO, using SQL Traversal to join the object to itself by its primary key. Note that with this technique, the records that reference the logical BO will still only be included in the export file once. At this point further subordinate instructions may use XPath notation to define the foreign key data. Using the physical BO as the primary instruction ensures that all records in the MO are considered. The subordinate instructions with the logical BO and XPath notations will only apply to the records that are applicable to that BO. This option is useful for MOs that have a small number of logical business objects with disparate foreign keys.
- Another option is to use the physical BO in the primary instruction and use raw SQLs in the subordinate instruction's traversal criteria to identify the foreign keys using substring commands. A separate Subordinate Instruction is needed for each SQL corresponding to each element occurrence. Using this technique has the same advantages of the previous in that all records for the MO are included in the migration. However, this technique may be useful for maintenance objects with a larger number of business objects expected where each has one or more foreign keys. It's especially useful if many business objects reference the same foreign key. Then only one instruction is required for that foreign key. Note that a single migration plan may use this technique and the XPath technique for different elements.

A migration request may have multiple migration plans for the same maintenance object. That allows for some flexibility and long term maintainability in that the above techniques may be used in multiple migration plans. Consider the following example:

- A product provides base business objects with foreign keys defined in the XML field and provides the appropriate migration plan with instructions. An implementation extends this business object or perhaps creates their own business object for the same maintenance object and includes different additional foreign keys in the XML. Rather than duplicating the base migration plan and adding additional instructions for the additional foreign keys, the implementation can create a second migration plan for the MO with the additional foreign


Copyright © 2026, Oracle and/or its affiliates.

keys defined. A migration request should be defined to include both migration plans. In this case if the implementation has only one custom BO, they can choose to use the custom BO as the primary instruction as described above in the first option.

# Defining a Migration Request

Migration Requests are used to define the data to be included in a migration. To view or define a migration request, navigate using **Admin**, and then **Implementation Tools**, and then **Migration Request**.

Use the **Migration Request Query** portal to search for an existing migration request. Once a migration plan is selected, you are brought to the maintenance portal to view and maintain the selected record.

There are three types or classes of migration request. The system provides a base business object for each along with a migration request class, which matches the business object. The subsequent sections provide more information about each class of migration request.

Note that all migration requests support defining a Category, which allows implementers to categorize the migration request.

In addition, all classes of migration request include the following zones:

- **Migration Request** This zone contains display-only information about the selected record. Please see the zone's help text for information about this zone's fields.
- **Referencing Migration Requests** This zone is only visible if the displayed migration request is included in a Group migration request. It lists each group migration request that includes it.

Other zones may appear for specific classes of migration requests. See the following sections for more information.

### Criteria-based

This type of migration request defines a set of entities to be logically included together as part of a migration. For each migration instruction, selection criteria is defined to indicate which records for each MO should be included. Selection may be defined using SQL, an algorithm, an entity tag or explicit primary key values.

- For selection using SQL Statement, refer to the embedded help for examples.
- For selection using Algorithm, the algorithms that are eligible to be plugged in here are those valid for the **Migration Request - Select** system event.
- For selection using Entity Tag, reference a tag name. The entities linked to the tag at export time would be included in the export file.
- For selection using Specific Keys, the primary key (1 through 5) must be explicitly specified. Multiple rows are allowed.

### Entity List

This type of migration request allows the user to choose explicit MO / prime keys. The MOs that are eligible are those that are configured with a **Default Migration Plan** option. Although the user is managing MO / PKs, the migration instructions are still defined with a migration plan. The system maps the migration instructions in a similar way to a `Criteria-based` migration requests that use a Specific Key selection type. Note however that it will create a separate migration instruction for each MO / PK combination. It does not try to group all PKs for the same MO / migration under one migration instruction.


Copyright © 2026, Oracle and/or its affiliates.

For this type of migration request, a user adds a migration request record with its description and other main information. Then, a special zone **Add Entities** is provided to find and select records based on a selected maintenance object and add to the migration request. The user is prompted to provide a reference and comments, if desired. If the category selected is one that requires a reference, then this will be validated. Alternatively, you may also use the Collect Entity zone to add specific entities as you review them on their respective maintenance portals.

When maintaining a migration request with existing entities, they are visible in the zone **Migration Request Entities** . This zone allows a user to remove the entity from the list.

#### Group

This type of migration request points to other migration requests. This allows you to define separate migration requests that represent logical groupings of migration plan instructions for ease of maintenance, but to combine all the separate migration requests into a single "grouped" migration request for streamlined export / import purposes.

The CMA export process will build an extract that includes the union of all the objects that qualify for the export and group them together based on their relationships.

## Identifying Tables to Exclude From Migrations

Some maintenance objects that are eligible to be migrated may include child tables that should not be included in the migration. For example, if an object includes log tables, the entries in the log should reflect the actions on the object in that system, and will be different between the source system and the target system. If you have a custom Maintenance Object that includes tables you don't wish to migrate (such as a log table), use the **Non-Migrated Table** option on the MO to specify this table. All child records for this table will also be ignored during migration.

Another use case to consider is a child "many-to-many" table that connects two administrative objects and exists in the maintenance object of both tables. The child table may be in both MOs for convenience sake, but it may be that one MO is considered more of a "driver" object and the other more of a subordinate. If you are doing a migration where you want to copy a subset of objects, you may want to only copy the driver object and its children and their data but not their children. For example, a To Do Type includes a collection of valid To Do Roles and in turn the To Do Role refers to its To Do Types. If an implementation wants to copy a single To Do and include all its related information, including its To Do Roles, it does not want the migration of each To Do Role to in turn copy all its To Do Types (and their data).

> **Note**
>
> The MO option must be set in both the Source and Target systems for a given MO.

## Configuring Custom Objects for Migration

During the implementation and extension of the product, new custom administrative maintenance objects may be introduced. If your implementation would like to migrate records in those maintenance objects using the Content Migration Assistant, additional steps must be performed, which are highlighted in the following sections.

#### Physical Business Object

As described in Understanding the BO Filtering Process, the migration plan requires a business object for its instruction. The business object is used to identify the records eligible for inclusion in the migration. Assuming your custom tables use one or more "logical" business


Copyright © 2026, Oracle and/or its affiliates.

objects for their processing, your implementation must decide if these business objects are appropriate for use by the migration plans, or if a physical BO is warranted. If so, create an appropriate physical BO.

### Review MO Option Configuration

The following points highlight maintenance object (MO) configuration that should be reviewed or updated to support CMA:

- If a physical BO was created (above), link it to the MO as an option using the appropriate option type.
- Be sure that your MO defines an appropriate FK Reference and includes an Option on the MO that identifies the FK Reference. This is used by various portals and zones for CMA when showing detail about records being imported into the target region. Also be sure that this FK reference defines an Information program.
- As described in Identifying Tables to Exclude From Migrations, an MO option is used to identify child tables for an MO that should never be included in a migration. If your custom maintenance object includes a standard Log table, than the recommendation is to list that table as an excluded table. Depending on the specific design of the maintenance object, there may be other child tables to define.

### Characteristic Type Configuration

The CMA import process will attempt to create a log record for any administrative object that includes a log table. If your implementation has introduced any custom administrative tables that you plan to include in a migration request and it includes a log table, you must, to ensure that the log creation is successful, add your log table as a valid characteristic entity to the characteristic type `F1-MGO` (Migration Object).

Navigate to Characteristic Type and select the characteristic type `F1-MGO`. Navigate to the **Characteristic Entity** tab and add a row to include the characteristic entity for your custom maintenance object's log table.

### Standard CMA Configuration

Create one or more migration plans for the new object, depending on the type of data in the maintenance object and the types of migrations you envision:

- If you have implemented only one "logical" business object used to define the data in the MO, then a single migration plan that references the this BO (or the maybe the MO's physical BO) is appropriate.
- If you have implemented more than one "logical" business object, would the data for multiple business objects get copied together? Then perhaps a single migration plan that references the MO's physical BO is appropriate.
- Are there additional foreign keys defined using mapXML in the business object(s) for the MO? If so, then it is recommended to include sub-instructions to define the links. At this point, if multiple "logical" BOs exist, your implementation may choose to define all the additional elements in the same migration plan or choose to define separate migration plans for each logical BO.
- Your implementation may decide to define more than one migration plan for the same type of record based on the types of migrations you plan to include. For example, you may decide to include a migration plan that copies only the records in this maintenance object. You may decide to define another migration plan that copies the records in this MO along with related records in another MO (for a special type of migration). Having said that, be sure to design the migration plan with reuse in mind to minimize maintenance efforts.


Copyright © 2026, Oracle and/or its affiliates.

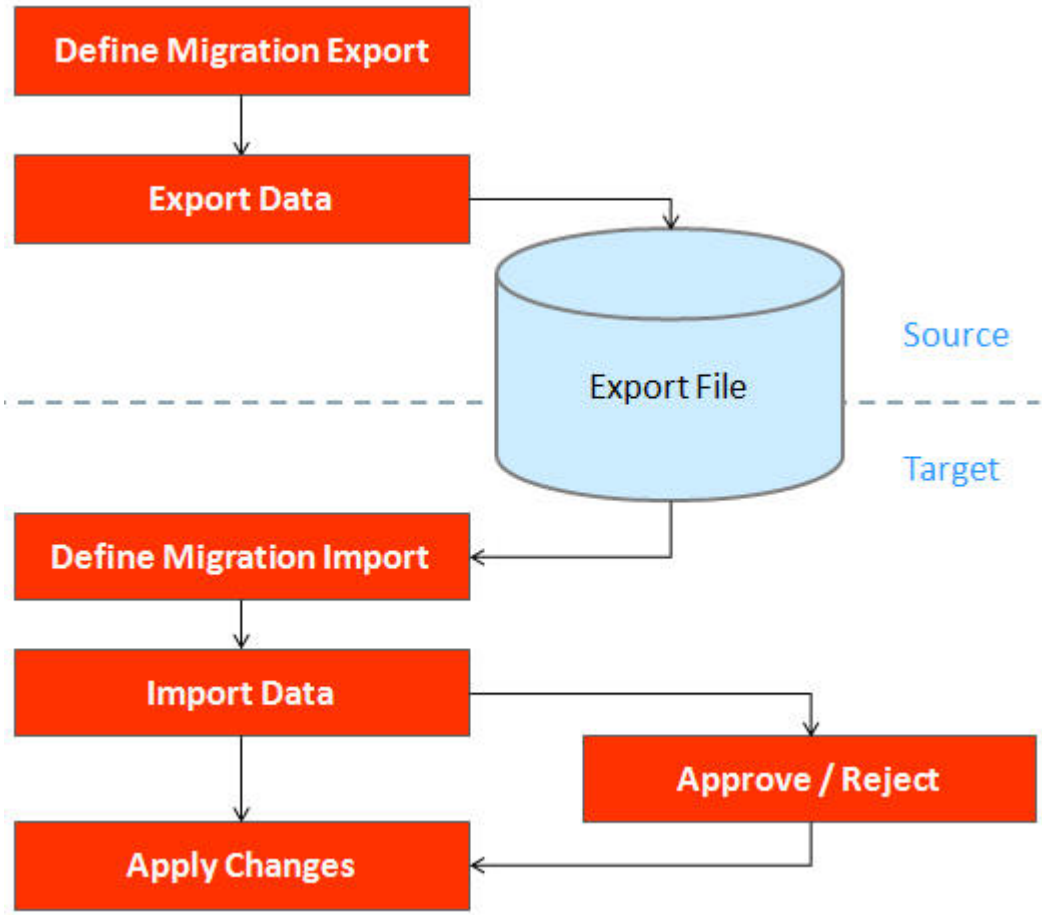
In order to support **Entity List** migration request, a default migration plan must be defined as an option on the maintenance object. This should be a single migration plan that supports all types of business objects for the MO.

If your implementation has a template migration request to use for migrations, include the new migration plan(s) as appropriate.

> ⚠ **Caution**
>
> Important! New migration plans and migration requests should follow naming conventions. Refer to System Data Naming Convention for more information.

# The CMA Execution Process

The following diagram illustrates a high-level view of the Content Migration Assistant execution process. The subprocesses illustrated here are described in more detail in the following sections.

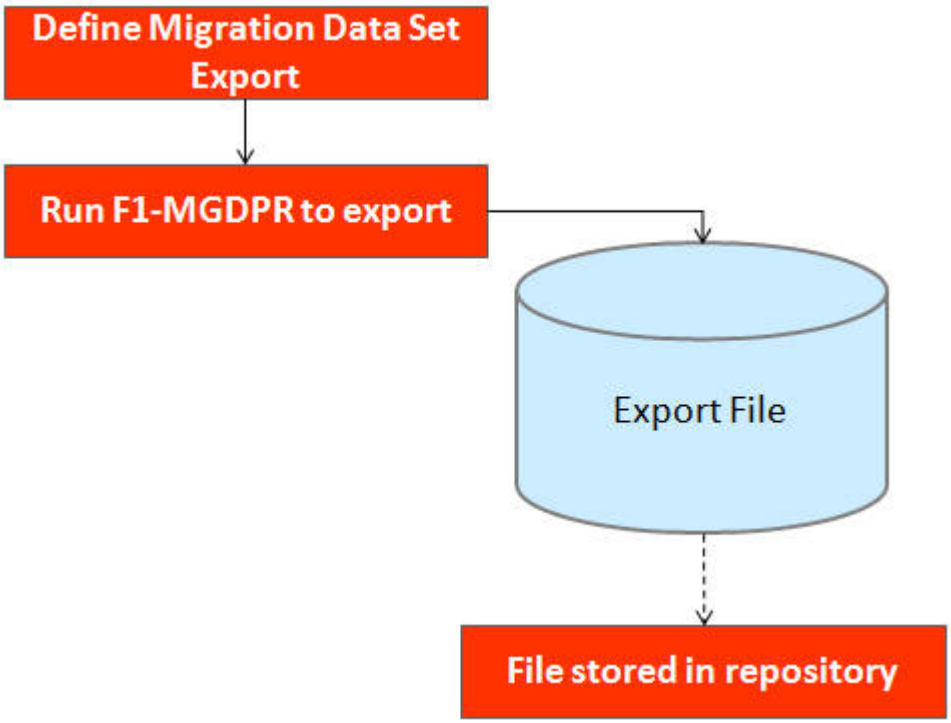



- Exporting a Migration
- Importing and Applying a Migration

## Exporting a Migration


Copyright © 2026, Oracle and/or its affiliates.

The migration export process begins in the source environment by defining a **Migration Data Set Export**, which specifies a defined **Migration Request** and provides a file name and description for the export file. After the data set is defined and saved, the **Migration Data Set Export Monitor** batch job can be submitted generate the export file. The lifecycle of the Migration Data Set Export business object includes algorithms that select the appropriate records according to the migration request, determine dependencies between records to build groupings of related objects and create the export file.




The following topics provide more detail about this process.

- Migration Data Set Export
- Export Lifecycle

## Migration Data Set Export

To migrate data from one region to another, define a **Migration Data Set Export**. This establishes the export file name and identifies the migration request.

To view an existing migration data set export, navigate using **Admin**, and then **Implementation Tools**, and then **Migration Data Set Export**. Use the query criteria to locate the desired data set.

Note that you can also initiate the creation of an export data set from the Migration Request portal using the **Export** button.

The export requires the name of an existing **Migration Request**.

Enter a unique **File Name** for the export. Do not use spaces in the name, and do not enter the file extension or a path. The output location and file extension of the intended export file, which should appear in the **Export Directory** and **File Suffix** labels, are defined as described in the topic Migration Assistant Configuration.


Copyright © 2026, Oracle and/or its affiliates.

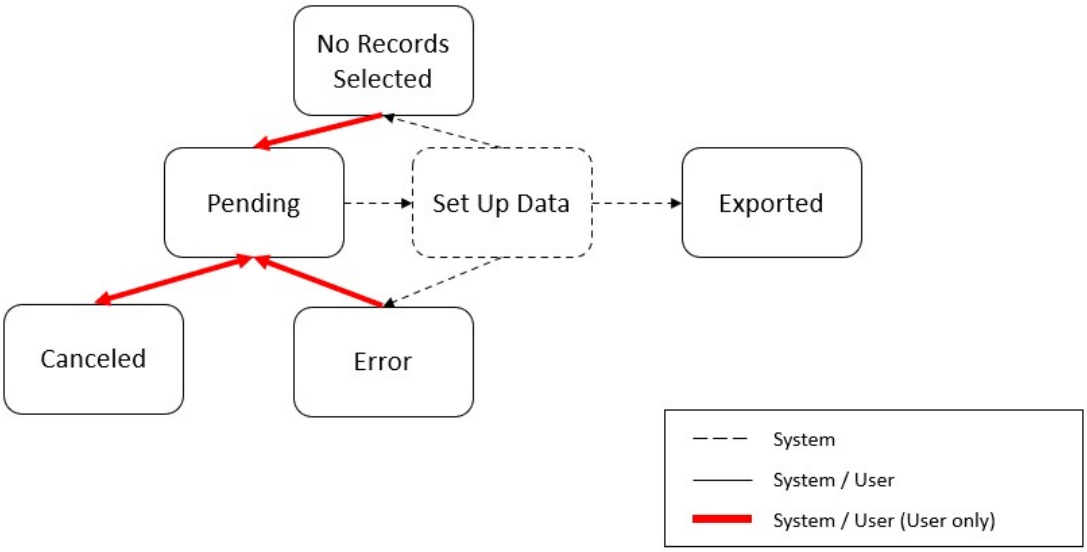
Enter an **Export Description** to provide information about the purpose of the export. Note that this field is not language enabled.

The **Source Environment Reference** is for information purposes. It should be populated with text that provides a meaningful description of the source environment. The default value is the URL of the source environment.

The **Migration File Version** is only visible if the product currently supports creating a file in an earlier file format version. If this element is visible and you are planning to import the file in a target environment that uses an older migration file format version, indicate the lower file format version that you wish the export to use.

## Export Lifecycle

The following diagram describes the lifecycle of a Migration Data Set Export (data set).




The following points describe the lifecycle.

- The data set is created in **Pending** status.
- A user may choose to temporarily **Cancel** a pending data set to prevent it from being processed. The user can later return it to the Pending state when desired.
- The record remains in Pending until its monitor batch job is submitted. The **Migration Data Set Export Monitor** (F1–MGDPR) selects pending records and transitions them to **Set up data**. Refer to Running Batch Jobs for more information.

> **Note**
>
> This batch control is delivered with the batch strategy configured as 'JOBS'. This ensures a more even distribution of work when there are multiple data sets to export.

- Set up data is a transitory state that includes the algorithm that does the work of determining the objects to include in the export and group related objects together into a transaction. If everything is successful, the export file is written to the appropriate file location and the record transitions to **Exported**. If the algorithm detects that no records are


Copyright © 2026, Oracle and/or its affiliates.

selected per the migration request instructions, the record transitions to **No Records Selected**. If an error is detected, the process stops and the record transitions to **Error**.

- If a record is in error and it is possible to correct the error, the record may be transitioned back to Pending to try again.
- If no records were selected and you want to adjust the migration request instructions to ensure that records are selected, the record may be transitioned back to Pending to try again.

When the process is marked as **Exported**, the export file can be imported into the target system.

> **Note**
>
> The export process creates a file, providing the benefits of having a standalone file. It can be stored in a version control system for audit purposes or provided to others for import purposes.

> **Caution**
>
> Under no circumstances should exported data files be edited manually. Doing so could cause data corruption when the file content is applied to the target environment.

> **Note**
>
> The export functionality is supported using the business object **Migration Data Set Export** (F1-MigrDataSetExport). The expectation is that implementations will use the delivered base business object and its logic and will have no reason to implement a custom business object for the CMA export process.

# Importing and Applying a Migration

The import process is broken down into four general steps: Import, Compare, Approve, Apply. The following points provide an overview of the steps.

- **Import**. The first step covers importing the file and creating appropriate Migration Import records in the target environment to facilitate the subsequent steps.
- **Compare**. The compare step reviews each object that is in the import file and compares the object in the import against the equivalent record in the target environment. The comparison step results in noting which objects are unchanged, which are new (and the appropriate SQL to insert them) and which objects are changed (and the appropriate SQL to update them). Based on user configuration at import time, the objects that qualify for the import may be in a state that requires review or may be pre-approved.
- **Approve**. Once the comparison is complete, the user should review the results. There may be records marked for review. All of these records must be approved or rejected before the import can proceed. Users may choose to suppress individual SQL statements for a given object that is approved. When the user is satisfied with the results of the comparison and has completed the review, the import is marked to proceed to the Apply step. Optionally, a migration import may be configured to automatically apply.


Copyright © 2026, Oracle and/or its affiliates.

- **Apply**. This is the final step and is the step where the records in the target environment are added or updated. Because of potential high volumes of data and because of possible dependencies between records, this step supports two levels of attempting to apply the records. There is an apply step at the object level and an apply step at the transaction level. This will be described in more detail below.


- Import Step
- Compare Step
- Approval Step
- Apply Step
- Adjusting Data Prior to Comparing
- Import Process Summary
- Cancelling a Data Set
- Additional Note Regarding Imports
- Caching Considerations
- Maintaining Import Data


## Import Step

The import process starts with verifying the import directory configured as described in the following topic Master Configuration - Migration Assistant and ensuring that the exported file is located in that directory. Then, in the target environment, a **Migration Data Set Import** record should be created. The user indicates the file name.

In addition, the user decides what the default status should be for resulting objects.

- The **Default Status for Add** sets the default status for objects that are determined to be ***new*** during the import comparison process. The default is to automatically set new objects to **Approved** status. Other options are to set any new objects to **Rejected** or **Needs Review** status.
- The **Default Status for Change** sets the default status for objects that are determined to be ***changed*** during the import comparison process. As with new objects, the default for changed objects is **Approved**, with **Rejected** or **Needs Review**options available.

The user may also configure the **Automatically Apply** flag to **Yes**. This allows for use cases where the migration is repetitive and has been tested and the user feels that there is no need for manual approval. Note that when configuring this setting, neither of the Default Status values may be set to **Needs Review** and at least one must be set to **Approved**.

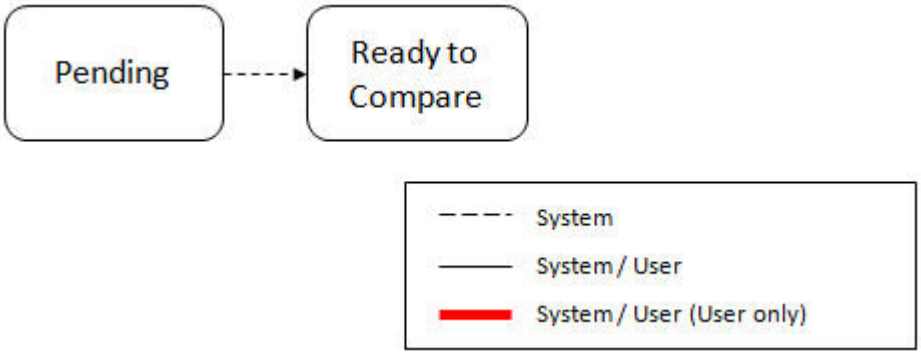
The file to import contains a list of all the objects included in the export. Any objects that the export step determined to be related have been grouped into “transactions”. Once the Migration Data Set Import is created, the next step is for the system to read in the file and create Migration Transactions and Migration Objects.

By default, the import step creates a Migration Object for each entity in the file. When importing a high volume data set of business entities, the user may request to import entities in bulk mode for performance reasons. In this mode, a single Migration Object record may contain multiple entities from the file. If at least one of the entities included in the migration object record is new and the remaining ones are either new or unchanged then the **Default Status for Add** setting applies to it. If at least one included entity is determined to be changed then the **Default Status for Change** applies to the record. Refer to bulk import mode for more information.



Copyright © 2026, Oracle and/or its affiliates.

The following is a portion of the Migration Data Set Import lifecycle as it pertains to the import step.



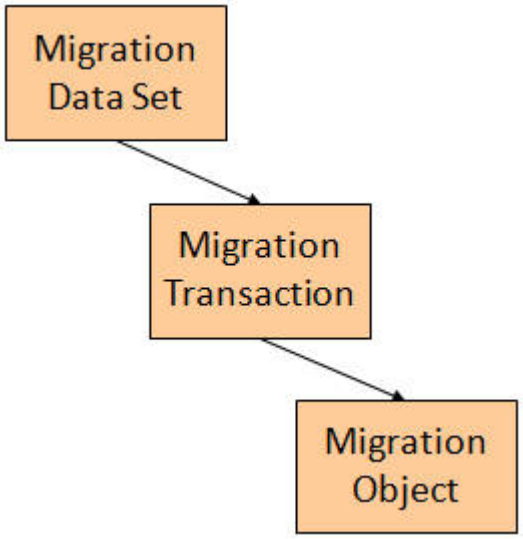

The following points describe the lifecycle.

- The data set is created in **Pending** status.
- The record remains in Pending until its monitor batch job is submitted. The **Migration Data Set Import Monitor** (F1–MGDIM) selects pending records and transitions them to **Ready to Compare**. Refer to Import Process Summary and Running Batch Jobs for more information.

  The Ready to Compare state has an algorithm that is responsible for reading the related import file and creating the migration transactions and migration objects. The data set remains in this state until the comparison step is complete.

> **Note**
>
> A user may choose to **Cancel** a data set. Refer to Cancelling a Data Set for more information.

The following diagram highlights the relationships of the resulting migration import records.




The migration transaction and migration object each have their own lifecycle that will help manage the subsequent compare and apply steps. At the end of the import step, the status values of the three types of records are as follows:

- Migration Data Set Import is in the **Ready to Compare** state.


Copyright © 2026, Oracle and/or its affiliates.

- Migration Transaction is in the **Pending** state.
- Migration Object is in the **Pending** state.

> **Note**
>
> The import functionality is supported using business objects supplied by the base product. The expectation is that implementations will use the delivered base business objects and their logic and will have no reason to implement a custom business objects for the CMA import process. The base business objects are **Migration Data Set Import** (F1-MigrObjectImport), **Migration Transaction** (F1-MigrTransactionImport) and **Migration Object** (F1-MigrObjectImport).

## Compare Step

The import step results in the creation of one or more migration objects. Typically, one migration record is created for each record selected in the export based on the export's migration request and its configuration. Related objects are grouped together in migration transactions.

When importing a high volume data set of business entities, the user may request to import entities in bulk mode for performance reasons. In this mode, a single Migration Object record may contain multiple entities from the file. Refer to bulk import mode for more information.

The next step in the import process is the Comparison step. In this step, the data captured by the import file for each object is compared to the view of that object in the target environment.

To cater for a possible large volume of objects, the comparison is done via a batch monitor. To aide in performance of the process, the monitor is performed on the migration objects so that it can be run multi-threaded. Once the objects are finished with the comparison, the migration transactions and the migration data set should be updated with an appropriate overall status before continuing to the next step. As a result, the comparison actually requires three steps: Migration Object Comparison, Migration Transaction Status Update and Migration Data Set Export Status Update. The steps are explained in detail in the following sections.

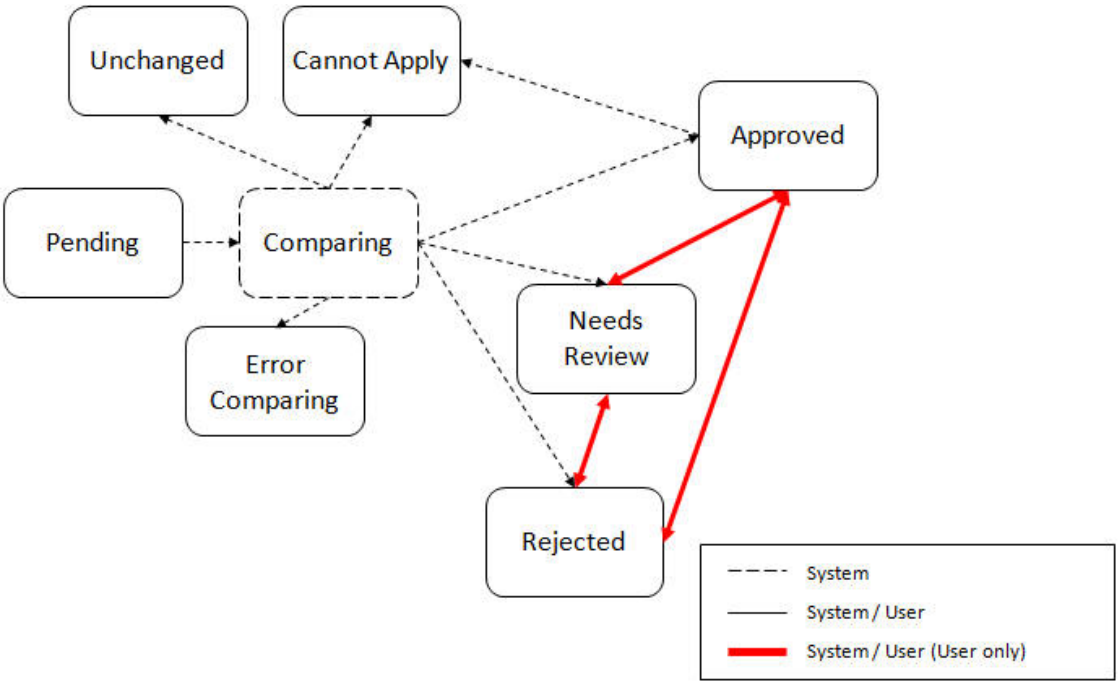
> **Note**
>
> Refer to Running Batch Jobs for more information about streamlining the various steps in the process.

### Migration Object Compare

This is the main step of the comparison. The **Migration Object Monitor** (F1–MGOPR) selects pending migration object records and transitions them to **Comparing**. This is a transitory state that includes the algorithm that does the work of comparing. There are various possible outcomes that could occur based on the logic in the algorithm. The following diagram illustrates a portion of the migration object lifecycle that pertains to comparison.

> **Note**
>
> When using separate batch processes for business data the **Migration Object Monitor (Business)** process (F1-MGOPB) works in the same way to compare business migration objects.


Copyright © 2026, Oracle and/or its affiliates.




The following points describe the lifecycle.

- When **Pending** records are selected by the monitor batch job, it transitions to **Comparing**. If the migration object refers to one or more pre-compare algorithms, they are executed to adjust the data prior to comparison. Then algorithm will determine the appropriate next state by comparing the source data to the target data.
- Entity related notes:
  - If the record in the migration object is found in the target environment and the data is exactly the same, the record transitions to **Unchanged** (with the object action value also set to **Unchanged**).
  - If the record in the migration object is found in the target environment and the data is different, the algorithm sets the object action value to **Change** and generates the appropriate SQL to be used later in the Apply step to update the record. It then transitions to **Approved**, **Needs Review** or **Rejected** based on the Default Status For Change setting captured on the Data Set.
  - If the record in the migration object is not found in the target environment, the algorithm sets the object action value to **Add** and generates the appropriate SQL to be used later in the Apply step to insert the record. It then transitions to **Approved**, **Needs Review** or **Rejected** based on the Default Status For Add setting captured on the Data Set.
  - When the import is performed in bulk mode each included entity is compared separately but the overall status of the record is set as follows:
    * If all included entities are unchanged then the migration object record transitions to **Unchanged** (with the object action value also set to **Unchanged**).
    * If all included entities are either unchanged or changed then the object action value is set to **Change**), corresponding SQLs are generated for the changed entities only and the migration object transitions based on the Default Status For Change setting captured on the Data Set.
    * If all included entities are either unchanged or new then the object action value is set to **Add**), corresponding SQLs are generated for the new entities only and the


Copyright © 2026, Oracle and/or its affiliates.

migration object transitions based on the Default Status For Add setting captured on the Data Set.

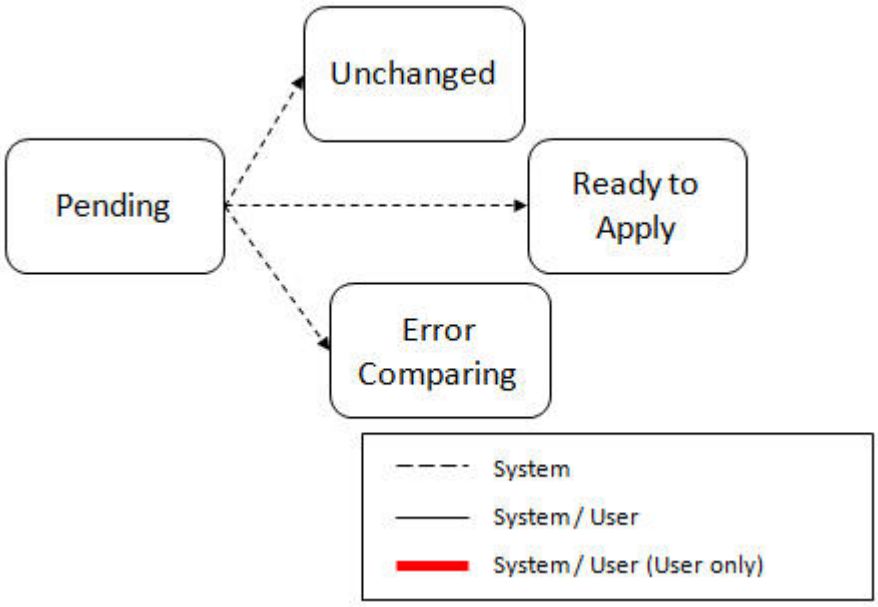
    * If some of the included entities are new and some changed then the object action value is set to **Mixed Actions**), corresponding SQLs are generated for the each entity and the migration object transitions based on the Default Status For Change setting captured on the Data Set.

- If there is any issue with attempting to parse the object data from the import, the record transitions to **Error Comparing**.
- If there is any reason that the imported object is not valid for import, the record transitions to **Cannot Apply**. The log will be updated with the error that caused the record to transition to this state. An example is that perhaps the record was exported in a different version of the product and has additional elements that are not recognized in this version.

> **Note**
>
> Refer to Cancelling a Data Set for information about cancelling a data set and its impact on its related objects.

### Migration Transaction Status Update

After the import step, the migration transaction remains in the Pending state until all its objects have completed the comparison step. At that point, the status of the transactions should be updated based on the results of their objects. The **Migration Transaction Monitor** (F1–MGTPR) selects pending migration transaction records and runs its monitor algorithms. There are various possible outcomes that could occur based on the logic in the algorithms. The following diagram illustrates a portion of the migration transaction lifecycle that pertains to comparison.




The following points describe the lifecycle possible next states after Pending.

- If any related migration object is in the Error Comparing state, the transaction transitions to **Error Comparing**.
- If all related migration objects are in the Unchanged state, the transaction transitions to **Unchanged**.


Copyright © 2026, Oracle and/or its affiliates.

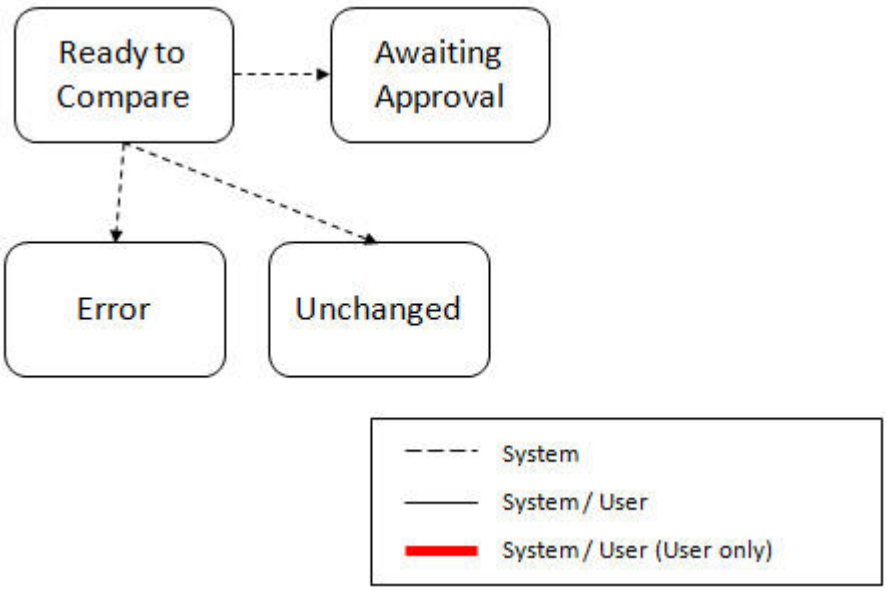
- Otherwise, the transaction transitions to **Ready to Apply**. This means that at least one object is in an “apply-able” state.

The transaction remains in the **Ready to Apply** state until a user has approved the data set to move to the Apply step and the transaction’s related objects have attempted to apply themselves. This is described in more detail below.

> **Note**
>
> Refer to Cancelling a Data Set for information about cancelling a data set and its impact on its related objects.

### Migration Data Set Import Status Update

Once all the objects and all transactions have been updated via the previous two steps, the migration data set export must be updated based on the results of their transactions. The **Migration Data Set Import Monitor** (F1-MGDIM) selects Ready to Compare data sets and runs its monitor algorithms. Note that this is the same monitor process that is used to select Pending data sets. There are various possible outcomes that could occur based on the logic in the algorithms. The following diagram illustrates the portion of the migration transaction lifecycle that pertains to comparison.




The following points describe the lifecycle possible next states after Ready to Compare.

- If any related migration transactions is in the Error Comparing state, the data set transitions to **Error**.
- If all related migration transactions are in the Unchanged state, the data set transitions to **Unchanged**.
- Otherwise, the transaction transitions to **Awaiting Approval**. This means that there are no errors and at least one object is in an “apply-able” state.

The data set remains in the **Awaiting Approval** state until a user decides that the data set and all its records are ready to progress to the Apply step.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> A user can choose to cancel a data set at any time while it is in progress. Refer to Cancelling a Data Set for more information.

## Approval Step

Once the comparison is complete and the data set transitions to the Awaiting Approval state, a user needs to progress the data set to **Apply Objects** to trigger the Apply step. The following points describe steps a user may take during the approval step.

- If the data set configuration for the Default State for Add and Change was set to **Approved**, then any migration object that is determined to be eligible for the Apply step will be in the Approved state. In this situation, a user may want to review the data set and its transactions and objects to see verify that the results make sense. At that time, the user is able to move an object to Needs Review or Rejected as appropriate.
- If the data set configuration for the Default State for Add and Change was set to **Needs Review** for either option, then each migration object in the Needs Review state must be reviewed and the user must either Reject or Approve each object before moving to the Apply step.
- If the data set configuration for the Default State for Add and Change was set to **Rejected** for either option, the assumption is that the rejected records don't need to be reviewed. But if a user finds a rejected record that shouldn't be rejected, it may be transitioned to Approved (or Needs Review) as appropriate.

A user has the option to mark one or more SQLs within a migration object to be suppressed. In this case, if the migration object is approved, then only the SQLs that are not marked as Suppressed will be applied. This option is useful when importing a record that has many child records. It may be that the user wants to only include a subset of the child records. This may be done by suppressing the child records that should not be included.

Once the user is comfortable with the data set's results and no more objects are in the Needs Review state, the user should transition the record to **Apply Objects**. This will initiate the Apply step.

Alternatively, if the Automatically Apply flag was set to **Yes** when creating the import record, the import data set will progress from **Awaiting Approval** to **Apply Objects**. Refer to Import Process Summary for more information.

> **Note**
>
> Refer to Maintaining Import Data for details about the pages provided to help the user review a data set and its transactions and objects to help in the approval step.

> **Note**
>
> A user can choose to cancel a data set at any time while it is in progress.

## Apply Step


Copyright © 2026, Oracle and/or its affiliates.

The apply step is the step where records in the target environment are added or updated. Like the comparison step, the apply step is actually multiple steps to optimally handle high volume and dependencies between records as smoothly as possible.

> **Note**
>
> Refer to Running Batch Jobs for more information about streamlining the various steps in the process.

Before explaining the apply steps in detail, the following points highlight the type of data that may be included in a given data set.

1. Records that have no foreign keys and therefore no dependencies on other records. Examples: Message, Display Profile.
2. Records that have foreign keys that may already be in the target. Examples: Algorithms for existing algorithm types, To Do Roles for existing To Do Types.
3. Records that have foreign keys that are new records but also part of the migration. CMA detected the relationship at export time and grouped the objects in the same transaction. Example: Script-based Algorithm Type where the script is also in the migration.
4. Records that have foreign keys that are new records but also part of the migration. CMA did not detect the relationship. This may occur if the referenced foreign key is in a XML or parameter column and the migration plan did not include an instruction to explicitly define the relationship. Example, a Zone that references a visibility script.
5. Records that have circular references where both records are new and are part of the migration. CMA detected the relationship at export time and grouped the objects in the same transaction. Example: plug-in Script for a BO plug-in spot. The script references the BO and the BO references an algorithm for the script's algorithm type. Another example is when the same record is maintained by multiple maintenance objects and therefore exists in multiple migration objects.

To handle high volume data, the first step in the apply process is to perform the apply logic at the migration object level via a multi-threaded batch job. This should result in all records in categories 1 and 2 above being applied successfully.

For records in categories 3 and 4 above, if a record with a foreign key is added or updated before its related record, the validation will fail. However, if the related record is added first and then the record referring to it is added, validation will pass. The tool handles these dependencies as follows:

- Dependency between master and transaction entities is typically hierarchical and in most cases straight-forward. The tool leverages that knowledge to orchestrate the processing of objects in an optimal way that follows their dependency order as much as possible. Note that relationship between entities could be complex and this approach does not eliminate all processing order related errors but rather significantly reduce them.
- Dependency between configuration entities is more complex and inter-twined and therefore migration objects are not ordered, i.e. the multi-threaded batch process may not process records in the desired order.
- To overcome the potential issue of processing order related errors, the Apply step has special functionality, described in detail below.

For records in category 5 above, the circular reference will mean that the apply process at the object level will not successfully add or update these records. The apply process at the transaction level will cover these records. This is described in detail below.


Copyright © 2026, Oracle and/or its affiliates.

### Apply Objects

Once the Data Set is in the state of **Apply Objects**, the **Migration Object Monitor - Apply** process (F1-MGOAP) runs to attempt to apply the objects.

> **Note**
>
> When using separate batch processes for business data the **Migration Object Monitor (Business)- Apply** process (F1-MGOAB) works in the same way to apply business migration objects.

The background process in conjunction with the Apply algorithm have special functionality to ensure records in categories 3 and 4 (above) successfully apply during this step:

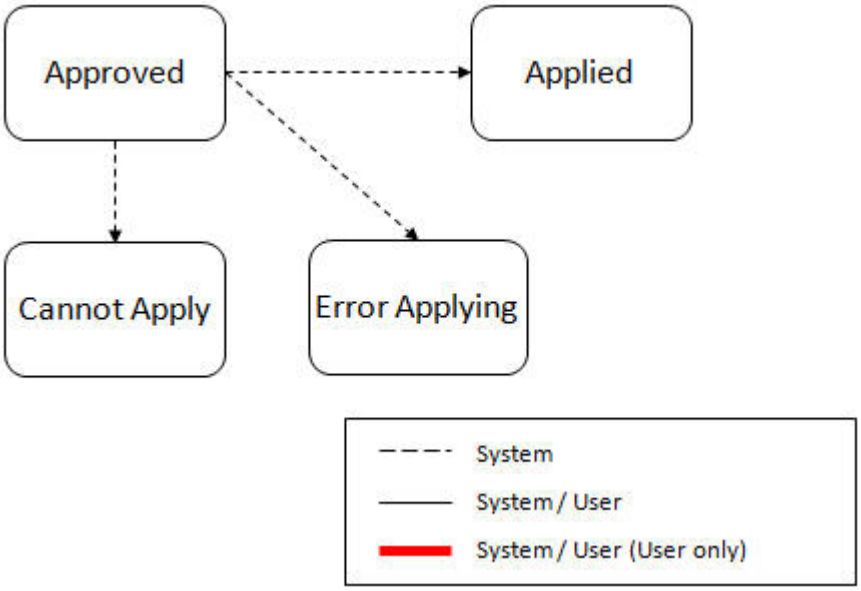
- The **Migration Object Monitor - Apply** process is a special one that continually re-selects records in the **Approved** state until there are no more eligible records to process.
- When an error is received in the Apply Object algorithm, the algorithm increments an "iteration count" on the migration object record. If the iteration count does not exceed a maximum count (noted in the algorithm), the object remains in the **Approved** state and is eligible to be picked up for processing again. If the iteration count exceeds the maximum defined in the algorithm, the record transitions to the **Error Applying** state.

> **Note**
>
> When submitting this Apply batch job, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. Doing this will cause the ‘excess’ threads to wait for the supported number of threads to finish.

> **Note**
>
> Each retry iteration to apply an object after an error slows down the Apply batch process. When the quality of the imported data is suspected to result in many errors, you may use the **Maximum Retries** batch parameter. Refer to the batch control's parameter description for more information.

The following diagram is the portion of the migration object lifecycle that pertains to the Apply step.


Copyright © 2026, Oracle and/or its affiliates.




At the completion of the Apply monitor process, typically the objects will be in the **Applied** state or the **Error Applying** state. The records in the Error Applying state are in that state for one of two reasons.

- They are in category 5 described above where the records have a circular reference with another record. For this scenario, the Apply Transactions step described below should successfully apply the records.
- There is some other error that is unrelated to the records in the current migration. In this case, manual intervention may be required. Refer to the Resolving Errors section below for more information.

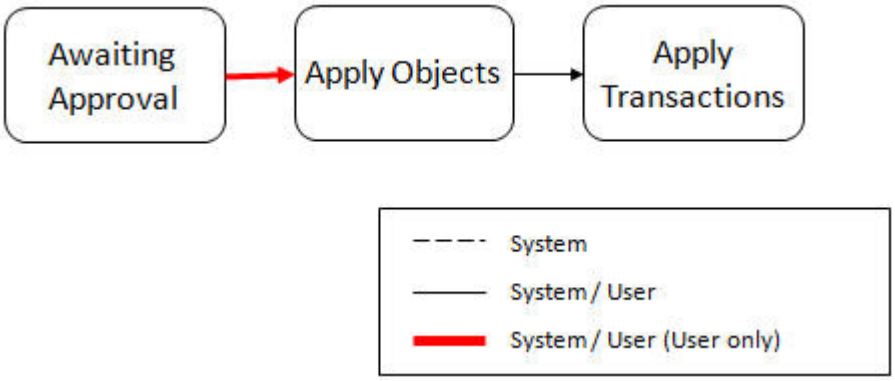
As shown in the diagram, the Apply Objects algorithm may also detect a reason that the object cannot be applied. This may occur if the object in the target environment has been updated since the comparison step, making the SQL captured at that point no longer applicable. If this occurs, after the current migration is fully applied, the original file may imported again, and new comparisons can be generated and applied.

### Apply Transactions

Ideally, after the Apply Objects step, all the objects are **Applied** or are in **Error Applying** due to the "circular reference" situation. The typical next step is to turn over responsibility to the transactions. The migration transactions can then attempt to apply their objects in bulk.

In order to ensure that multiple background processes are not trying to select migration objects to run the Apply step, the Transactions are only eligible to attempt to "apply my objects" if the Data Set is in the **Apply Transactions** state.


Copyright © 2026, Oracle and/or its affiliates.




A monitor algorithm (executed by the data set monitor batch process) on the Apply Objects state checks to see if all migration objects are no longer **Approved** or the count of records in **Error Applying** does not exceed a configured limit. If so, it automatically transitions the record to the **Apply Transactions** state.

If the number of objects in **Error Applying** exceeds a configured limit, the monitor algorithm does not automatically transition the record. In that case, a user must determine if the large number of errors can be resolved or manually transition to **Apply Transactions** (despite the large number of errors). The Resolving Errors section below describes alternative steps that the user may take if there are errors.

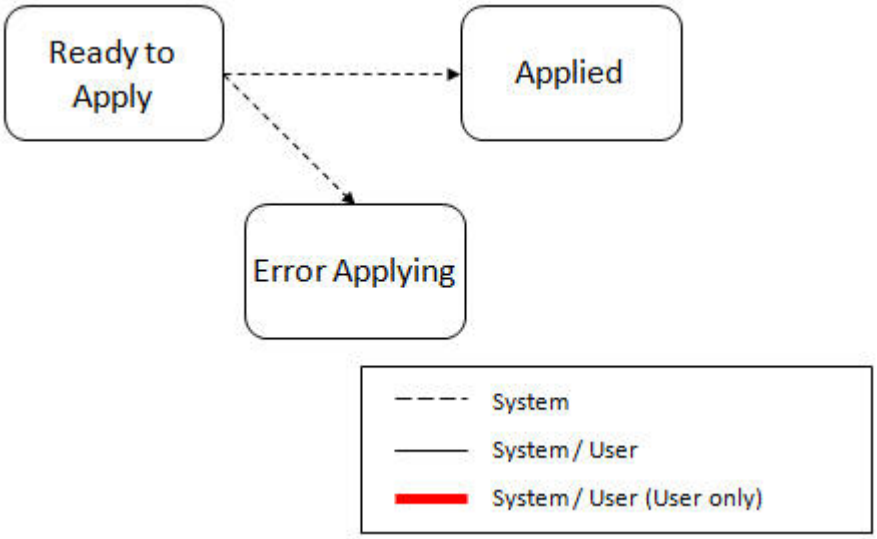
Once the Data Set is in the state of **Apply Transactions**, the **Migration Transaction Monitor - Apply** process (F1-MGTAP) runs. It attempts to apply the transaction's objects. If no migration objects are in error, the migration transaction simply transitions to **Applied**. If any of the migration objects are in **Error Applying**, the background process and the Apply algorithm have special functionality to try to overcome dependencies in migrated objects:

- The Apply algorithm selects all migration objects in error and performs all their SQL, then validates all the records. If there are objects in the transaction with circular references, they should pass validation at this point.
- Because there may still be some dependencies across transactions, similar error handling described in the Apply Objects step occurs here. When an error is received in the Apply Transaction's Object algorithm for any of the objects in the transaction, the algorithm increments an "iteration count" on the migration transaction record. If the iteration count does not exceed a maximum count (noted in the algorithm), the transaction remains in the **Ready to Apply** state and is eligible to be picked up for processing again. If the iteration count exceeds the maximum, the record transitions to the **Error Applying** state. Note that if any objects in the transaction are in error, none of the objects are applied. They all remain in error.
- The **Migration Transaction Monitor - Apply** process is a special one that continually re-selects records in the **Ready to Apply** state until there are no more eligible records to process.

> **Note**
>
> When submitting this Apply batch job, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. Doing this will cause the 'excess' threads to wait for the supported number of threads to finish, erasing the benefit of the iteration processing.

The following diagram is the portion of the migration transaction lifecycle that pertains to the Apply step illustrating the points above.


Copyright © 2026, Oracle and/or its affiliates.




If at the end of the transaction level Apply process there are transactions in error (and therefore there are still objects in error), a user must review the errors and determine how to fix them. Refer to the Resolving Errors section below for more information.

### Resolving Errors

As mentioned in the previous sections, errors may be received after the Apply Objects process runs. If the number of records in error is below a certain limit (and the data set monitor batch job is submitted to execute the monitor algorithms) the system will automatically transition the data set to the **Apply Transactions**. If the monitor batch job is not run or if the number of objects in error exceeds a certain limit, a user must make the decision after viewing the errors in the Objects in Error zone on the Migration Data Set Import portal.

- If the errors appear to be dependency related, the user can decide to let the "transactions apply their objects" and transition the data set to **Apply Transactions**, described above.
- If the errors appear to be related to an outside issue that can be manually resolved, the user may choose to fix the issue and redo the Apply Objects step.
- The user may also decide to reject one or more objects to remove them from the migration.

After the Apply Transactions step, if there are still errors, a user must review the records and determine how to proceed. Errors are visible in the **Transactions in Error** zone on the Migration Data Set Import portal.

- The user may decide to reject one or more objects to remove them from the migration.
- The user may manually resolve an issue external to the migration and then decide to do one of the following:
  - Redo the **Apply Objects** step. This is recommended if there are a large number of Objects still in error and not a large number of dependencies expected. The benefits of running the Apply Objects multi-threaded will ensure that the process runs efficiently.
  - Redo the **Apply Transactions** step.

Because the objects and transactions are in Error Applying, in order to "retry" the Apply step after manually fixing an error, the system needs to move the records back to the state that allows them to be picked up by the appropriate Apply process. For migration objects, records need to be moved back to **Approved**. For migration transactions, records need to be moved back to **Ready to Apply**. The following points describe the Retry logic for migration objects.


Copyright © 2026, Oracle and/or its affiliates.

- If a user decides to **Retry Objects** (using an action button on the Migration Data Set Import page), the data set transitions to the **Retry Objects** state. At this point the Migration Object monitor must be run.
- The monitor on the **Error Applying** state for the objects detects that the data set is in the state of **Retry Objects** and that triggers the transition back to **Approved**.
- The next step is to transition the data set from **Retry Objects** to **Apply Objects**. This may be done manually or by running the Migration Data Set Import monitor process.
- Now the objects are eligible to be picked up by the object level Apply process.

Analogous logic exists for the migration transactions.

- If a user decides to **Retry Transactions** (using an action button on the Migration Data Set Import page), the data set transitions to the **Retry Transactions** state. At this point the Migration Transaction monitor must be run.
- The monitor on the **Error Applying** state for the transactions detects that the data set is in the state of **Retry Transactions** and that triggers the transition back to **Ready to Apply**.
- The next step is to transition the data set from **Retry Transactions** to **Apply Transactions**. This may be done manually or by running the Migration Data Set Import monitor process.
- Now the transactions are eligible to be picked up by the transaction level Apply process.

The retry logic may also occur when transitioning between the Apply Objects and Apply Transactions depending on whether or not there are errors. The following scenario highlights this point.

- After the **Apply Objects**step there are objects in **Error Applying**. The data set transitions to **Apply Transactions** and the Apply step is done at the transaction level.
- After the **Apply Transactions** step there are transactions in **Error Applying**.
- User chooses to try to apply objects again (by clicking **Retry Objects**). The steps outlined above for retrying objects are followed at this point.
- After the apply objects, user may choose to retry objects again (after fixing errors if applicable).
- At some point the user will transition to**Apply Transactions** again. If there are transactions in **Error Applying**, the system will automatically transition the data set to **Retry Transactions** and the steps outlined above for retry transactions are followed.

### Finalize Apply Step

Once all the migration objects for a migration transaction are in a final state (**Applied**, **Rejected** or **Cannot Apply**), the migration transaction transitions to the **Applied** state. Once all the migration transactions are in the **Applied** state, the Migration Data Set record transitions to the **Completed** and the import is complete.

> **Note**
>
> To review the full lifecycle for each record, refer to the Business Object - Summary tab in the application metadata for the base business objects **Migration Data Set Import** (F1-MigrObjectImport), **Migration Transaction** (F1-MigrTransactionImport) and **Migration Object** (F1-MigrObjectImport).


Copyright © 2026, Oracle and/or its affiliates.

## Adjusting Data Prior to Comparing

There are reasons that the data you are importing requires some adjustment before applying it in the target environment. In such cases, an algorithm plugged into the migration plan primary instruction may be used to adjust the data when importing. This algorithm is executed by the comparison algorithm before any comparison is performed. Algorithms of this system event receive the view of the source record (being imported) and the view of the existing record in the target region, if it exists. The data is provided using the physical BO of the migration plan's maintenance object. The algorithm may make changes and pass a new view of the record that should be used for the comparison. This system event supports multiple algorithms that are executed in sequence. Each algorithm receives the original record's data, the target record's data (if applicable) and the 'new' view of the data (as populated by previous algorithms, if any). The final 'new' view of the data is used for the object comparison. The system includes the algorithms linked to the primary instruction of the migration plan from the source environment. If the system finds any additional pre-compare algorithms linked to the migration plan's primary instruction in the target system, those algorithms are also executed.

> **NOT_SUPPORTED**
>
> Refer to Base Business Objects for more information about physical BOs.

Some examples of records that may require import algorithms.

- Batch Control references its next batch sequence number along with snapshot information like the last run date / time. This information is only relevant with respect to its environment. The instruction for a batch control can include an algorithm to not overwrite the batch sequence number when copying a batch control.
- Some products include administrative objects that reference a master data object. Master data objects are not copied as part of CMA. An import algorithm may be used to adjust the referenced master data foreign key when importing, for example to reset it (or not overwrite when updating). If the algorithm knows how to find the appropriate master data record to link, that may also be included.
- If you are importing data that from a source environment on a lower version, if the object you are importing has been enhanced in the latest release with a new required field, you may need to provide an algorithm to populate data in that field.

Note that it is possible to use the algorithm to "reset" the source data as a way of indicating that the record should not be imported. For these situations, the migration object comparison step will transition the record to **Unchanged** and will use an object action value of **Canceled**. (Note that object action is a simple lookup value. The record is not transitioned to the **Canceled** BO state as to reserve that status for user initiated cancellations of the object or one of its parent records). This technique not expected to be used often because ideally using appropriate selection criteria at export time should ensure that the only records exported are those that should be imported.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> **Legacy 'Import' system event.** The system originally provided an Import system event / plug-in spot. The purpose of algorithms for this plug-in spot were similar in that they were meant to adjust imported data prior to adding or updating. The algorithms were executed in the Apply step. The logic does not allow for easily interacting with the record using a BO. This makes it difficult to use a plug-in script as the plug-in type. In addition, it is difficult to update elements in an XML column. The support for the plug-in spot will be removed in a future release. Algorithms to adjust the data should be using the pre-compare system event.

## Import Process Summary

The following table summarizes the steps required to complete the import process from start to finish. Note that this section **only a summary** and assumes that you are familiar with the details described in the previous sections. It highlights what steps are manual and what steps are performed by a batch monitor process. For each step, the table highlights the Next Action sequence that would occur. For the Apply steps, there are two parts where multiple next actions are possible based on whether there are errors and the user's decision on how to resolve the error. Refer to Resolving Errors for more information. The possible next actions have the same sequence with a letter following the sequence highlighting the action to take based on the results of the previous step.

> **Note**
>
> When running the Apply batch jobs, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker.

Also note that a sequence and action marked in bold is considered the "normal path".

When using separate batch processes for business data any reference to a migration object monitor batch process applies also to the business data related batch controls.


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Load-Amount Actions | Next Action Sequence |
|---|---|---|---|---|---|
| Import | **10** | **Create Import record** | Manual | Mig Rat ion Data Set Import - Adding | 11 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Next Action Sequence |
|---|---|---|---|---|
| | 11 | Import file | Batch | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Load-Amount Transactioned | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | Completed - Resulting Status | |
| | | | | Completed - Migration Transaction - Pe | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Pending - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | nding Migration Object - Pending | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Manual / Batch Action | Next Action Sequence |
|---|---|---|---|
| Compare | 20 | Batch Migration Object Compare | 21 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Portal-Action | Record Compacted-Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|---|
| | | | | or [WPen using separate business da | Review Rejected Unchanged or Err | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to ar LCD-Amortraction ed - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | to ar Process ensg submitals o F1-MGOPB( | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Tracked | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | Migration Object Monitor (Business | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Actioned - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | s)) ] | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Action Sequence | Next Action Sequence |
|---|---|---|---|---|---|
| | 21 | Migration Transaction status upda | Batch | FM 21-MGT Pro-Mirgamastaic Tomar Reactivity on AMP | 22 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | Resulting Status | |
| | | | | Priority, Unchanged or Error Comparing | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Procedure Action Code | Next Action Sequence |
|---|---|---|---|---|---|
| | 22 | Migration Data Set Import status up | Batch | FM-1-MGD-Mig-Data-Set-Import-Status | 30 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | date | | Map prior to overall, Apply Objects (if config | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Perform Action Control - Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | ured for Automatic Apply), Unchang | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control - Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | ed or Error | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Action Sequence | Next Action Sequence |
|---|---|---|---|---|---|
| Approval | **30** | **Review comparison results, approv** | Manual | Mid-Amount Approved, Reject Line | 40 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked | Next Action Sequence |
|---|---|---|---|---|---|
| | | **e/rejectasneeded** (This step is skip | | to the Tnrarnescaocrtisonhso/uOlbdjbeecitnsNaesendesc | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Portal-Action | Record Completed - Resulting Status | Next Action Sequence |
| --- | --- | --- | --- | --- | --- | --- |
| | | pped if the dataset is configured for | | essary) | eview)<br>· Migration Data Set Import - A | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Manual / Batch Action | Next Action Sequence |
|---|---|---|---|
| | | Automatic Apply) | Apply Objects |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Actioned - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| Apply | 40 | Apply Objects | Batch | F41 - Mgr Ca O Ap - Mobg reactti- Ap Objie ed to Mo Er ir to or | Appropriate next action is based on error review, if applicable. |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Action Attached - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | Apply [When using separate business | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Tracked | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | Processes Submitals of 1-M | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amortization | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | GOAB (Migration Object Monitor (Bu | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Actioned - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | siness) - Apply)] | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Procedure Action Code | Next Action Sequence |
|---|---|---|---|---|---|
| | 41a | Migration Data Set Imports status up | Batch | FM-1-MGD-Mig-Data Set Import Status Up | 42 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Contract Code | Next Action Sequence |
|---|---|---|---|---|---|
| | | date-auto transition to Apply Tran | | Manual Actions | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | sactions. Only applicable if the nu | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Reached - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | mber of migration objects in Error A | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control - Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | pplying is below a threshold | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Actioned-Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | **41b** | **Migration Data Set Import status up** | Manual | Migration Data Set Import-click Ap | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Reduced - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | **date-manual transition to Apply Tr** | | ply Transactions | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | **ansactions**. Occurs if user reviews | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Actioned - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | errors and determines that they may | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Trace Ordered - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | be resolved in the Apply Transactio | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | n s t e p . | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Parent to Child-Amount Tracked | Next Action Sequence |
|---|---|---|---|---|---|
| | | System detects that all the transac | | Migration Data Set Import - Apply Tr | 42 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Tracking Stated - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | tions are in the Ready to Apply state | | ansactions | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Tracked | Next Action Sequence |
|---|---|---|---|---|---|
| | | and proceeds to the Apply Transacti | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | onsstate. | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Manual / Batch Action | Next Action Sequence |
|---|---|---|---|
| | | System detects that there are trans | Migration Data Set Import - Retry Tr |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Perform Action Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | actions in the Error Applying state | | ansactions | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | and transitions instead to the Retr | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracking Stated - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | y Transactions state. | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Action Sequence | Next Action Sequence |
|---|---|---|---|---|---|
| | 41c | Migration Data Set Import status up | Manual | Migration Data Set Import - Clerk Web | 44 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Perform Action Code | Next Action Sequence |
|---|---|---|---|---|---|
| | | date-manual transition to Retry Ob | | trycObjects | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Accounted - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | jects. Occurs if user review error | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | sand decides to fix an external error | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | and wants to try the batch level Appl | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | yagain at the object level. | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Next Action Sequence |
|---|---|---|---|
| | 42 | Apply Transactions | Appropriate next action is based on error review, if applicable. |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Load-Amortization | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | Error to or Applying Migration Object - A | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Perform Action Control | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | Amount-Adjusted-Resulting Status | |
| | | | | Applied or Error Applying | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Next Action Sequence - Resulting Status |
|---|---|---|---|
| | 43a | Manual / Batch<br>Migration Data Set Import status up | N/A<br>Migration Data Set Import status up |

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | date-autotransition to Completed | | Monitor | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | Applicable if all transactions are | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | Applied | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Action Sequence | Next Action Sequence |
|---|---|---|---|---|---|
| | 43b | Migration Data Set Import status up | Manual | Migration Data Set Import - Clerk Web | 44 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracking Stated - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | date-manual transition to Retry Ob | tryc Objects | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | jects. Occurs if user review error | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracking Stated - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | sand decides to fix an external error | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracking Method - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | and wants to try the batch level Appl | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | yagainatthe Objectlevel. | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Next Action Sequence |
|---|---|---|---|
| | 43c | Manual<br>Migration Data Set Import status up | Manual<br>Migration Data Set Import status up |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Perform Action Code | Next Action Sequence |
|---|---|---|---|---|---|
| | | date-manual transition to Retry Tr | | Perform Action Code - Amount Transaction Detailed - Resulting Status | any Transactions |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | ansactions. Occurs if user reviews | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control - Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | errors and decides to fix an external | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control - Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | error and wants to try the batch leve | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Tracking Stated - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | l Apply again at the Transaction lev | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | el. | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Action Sequence | Next Action Sequence |
|---|---|---|---|---|---|
| | 44 | Migration Objects status update fr | Batch | Fi-gMCaOPRo-nMibgreactio-ApOprjoevcetdMonit | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Reached - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | om Error Applying back to Approved. | or [When using separate business da | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | Occurs if user chose to Retry Object | | taprocesses submitals o F1-MGOPB( | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control - Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | s. | Migration Object Monitor (Business | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Manual / Batch Action | Next Action Sequence |
|---|---|---|---|
| | | | Reactor Cold-Amorphic Ordered-Resulting Status s)) ] |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Contract Code | Next Action Sequence |
|---|---|---|---|---|---|
| | | | Batch | FM-Migration Data Set Import Status Update | 40 |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process Action Sequence | Next Action Sequence |
|---|---|---|---|---|---|
| | | date from Retry Objects to Apply Obj | | Manual - Resulting Status | Monitor |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | ects | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Next Action Sequence |
|---|---|---|---|
| | 45 | Batch Integration Transactions status update | Manual / Batch |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Reject to Error | Next Action Sequence | Resulting Status |
|---|---|---|---|---|---|---|
| | | after Error Applying back to Read | | | | Supervisor |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Apply-Occurs if user chose to Re | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control-Amount Tracking Stated - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | try Transaction so rif the user tran | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Apply Transactions and Credit Amount | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | hesystemdetectsthattherearerTra | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | nsactionsinthe Error Applyingsta | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Parent to Local-Amount Tracking | Next Action Sequence |
|---|---|---|---|---|---|
| | | | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Next Action Sequence - Resulting Status |
|---|---|---|---|
| | | Batch Migration Data Set Import status up | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | date from Retry Objects to Apply Obj | | Mosaic to Ions | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Seq | Action | Manual / Batch | Process to Control Amount Tracked - Resulting Status | Next Action Sequence |
|---|---|---|---|---|---|
| | | | ects | | |

The following table summarizes the batch monitor jobs that are used in the import process. You can see that there are special monitor processes for the Apply step for both the Object and Transaction. However, for all other states that have monitor logic, the standard monitor process for that MO is used.

| Batch Control | Description | Comments |
|---|---|---|
| F1-MGDIM | Migration Data Set Import Monitor | Processes data set records in the following states:<br>• Pending<br>• Ready to Compare<br>• Apply Objects<br>• Retry Objects<br>• Apply Transactions<br>• Retry Transactions |
| F1-MGTPR | Migration Transaction Monitor (Deferred) | Processes transaction records in the following states:<br>• Pending<br>• Error Applying |


Copyright © 2026, Oracle and/or its affiliates.

| Batch Control | Description | Comments |
| --- | --- | --- |
| F1-MGTAP | Migration Transaction Monitor - Apply | Processes transaction records in the Ready to Apply state where the data set is in the Apply Transactions or Canceled state.<br><br>**Note**<br>Be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. |
| F1-MGOPR | Migration Object Monitor | Processes object records in the following states:<br>• Pending<br>• Needs Review (check for Data Set cancellation)<br>• Error Applying |
| F1-MGOPB | Migration Object Monitor (Business)<br>Applicable only when separate business data processing is enabled. | Similar to F1-MGOPR (Migration Object Monitor) but restricted to business data migration objects. |


Copyright © 2026, Oracle and/or its affiliates.

| Batch Control | Description | Comments |
|---|---|---|
| F1-MGOAP | Migration Object Monitor - Apply | Processes object records in the Approved state where the data set is in the Apply Objects or Canceled state.<br><br>**Note**<br>Be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. |
| F1-MGOAB | Migration Object Monitor (Business) - Apply<br>Applicable only when separate business data processing is enabled. | Similar to F1-MGOAP (Migration Object Monitor - Apply) but restricted to business data migration objects. |

Refer to Running Batch Jobs for more information about managing the batch jobs, including ways to automate the above steps.

## Cancelling a Data Set

A user may choose to **Cancel** a data set to prevent it from being processed at any point during the process.

The request sets the data set in **Pending Cancel** status. If related migration objects have not yet been created, the data set transitions immediately to the **Cancelled** status and the cancellation request is complete.

If related migration transactions or migration objects have already been created, they will not be canceled as part of the data set getting canceled (due to possible high volumes of related records). They will be canceled the next time an appropriate monitor batch process runs. The child records checks to see if the data set has been canceled prior to any state transition.

Once all the migration transactions and objects are in a final state, the next batch monitor processing of the data set transitions it to the **Cancelled** status to complete the cancellation request.


Copyright © 2026, Oracle and/or its affiliates.

## Additional Note Regarding Imports

The following points describe miscellaneous comments related to Migration Import.

- CMA relies on the fact that database referential integrity constraints are not in place, and that the SQL statements can be run in any order within the transaction. Any archiving solution that requires referential integrity constraints (such as Information Lifecycle Management) would not be possible on this data. Given that CMA migrations comprise administrative data and not transactional data, this should be a reasonable exception.
- The validation that is performed is only via the **Page Validate** service. BO validation algorithms are not executed. Page validation does not include validation of the business object against the schema (for example, for required fields, field sizes, etc.).
- If multiple migration requests are exported at the same time, on the import side, you should consider importing, reviewing, and applying an entire file/data set before moving on to the next one. The reason is that if objects are included in more than one file, two sets of "inserts" will be generated, but only the first will succeed. The second will cause the object to transition to "Cannot Apply". If instead you wait until the first file is completed before importing the second, the second data set will not generate any SQL for the object, since it has already been inserted. It's a matter of efficiency: If you first import all files and then try to apply all, you'll have to identify the duplicated object as an error and then mark the object as rejected before applying the transaction. This may also be avoided by using a Group migration request to include all objects in one file rather than multiple files.
- The system provides an algorithm to purge "unchanged" migration objects for a given migration data set. This may be plugged in as a BO exit algorithm on the **Ready to Compare** state for the Migration Data Set Import business object (`F1-MigrDataSetImport`).

The following points describe miscellaneous comments related to importing a business data set in bulk mode.

- The number of entities included in each migration object is captured on the migration object record.
- The comparison step only generates SQLs for new or changed entities. SQLs are not generated for unchanged entities.
- Searching for migration objects based on their included entities is only supported for entities that were determined as new or changed. In other words, this type of search by included entities is only possible after the comparison step is performed.

## Caching Considerations

There are two important caching considerations when using CMA to migrate administrative data.

- The "apply" batch processes update data. As described in Batch Cache, if any batch job is updating data marked as "cached for batch" it must be run in a special thread pool. Contact your system administrator and ask for the name of a thread pool that has turned off caching. This thread pool name must be entered in the batch controls used for CMA processing in the **Thread Pool Name** parameter.
- After a successful migration, the target region now has new administrative data which needs to be part of various caches. It is recommended to flush the server cache (which will trigger a 'global' flush of the cache). If the thread pool workers in the target region are configured to refresh their caches when a global flush is requested, then this is the only


Copyright © 2026, Oracle and/or its affiliates.

step required. If not, then the F1-FLUSH batch job should also be submitted to refresh the caches used in batch processing.

**NOT_SUPPORTED**

Refer to Caching Overview for more information.

**NOT_SUPPORTED**

Refer to Running Batch Jobs for more information about running CMA batch jobs.

# Maintaining Import Data

This section describes the portals provided to add, view and maintain migration import data.

- Migration Data Set Import
- Migration Transaction Portal
- Migration Object Portal
- Configuration Deletion

## Migration Data Set Import

Use the Migration Data Set Import portal to view and maintain migration data set import records. Refer to Importing and Applying a Migration for an overview of the import process.

Navigate using **Admin**, and then **Implementation Tools**, and then **Migration Data Set Import.** You are brought to a query portal with options for searching for import data sets. In addition, the query provides an option to specifically search for data sets that have either objects in error or transactions in error. Once a data set has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones are visible on the **Main** tab:

- **Migration Data Set Import.** This zone contains display-only information about the selected record. Please see the zone's help text for information about this zone's fields.
- **Entity Summary.** This zone provides a summary count of all migration objects included in the current data set across the various statuses.

The following zones are visible on the **Transactions** tab:

- **Migration Data Set Transactions in Error.** This zone is only visible if there are transactions for this data set in a non-final state that have errors. It indicates the error for each transaction. A user may use this zone to review errors after the monitor batch job to apply transactions completes. The errors received when attempting to apply objects at the transaction level may differ from those received when attempting to apply objects at the object level. A transaction log is created for each object error received and these exceptions are shown in this zone.

**Note**

Refer to Apply Step for more information about resolving errors.


Copyright © 2026, Oracle and/or its affiliates.

- **Migration Data Set Transactions.** This zone is visible once the Import Step has occurred and lists all the transactions that are related to the data set. To see more information about a specific migration transaction, click the hypertext for its ID. This brings you to the Migration Transaction portal.

The following zones are visible on the **Objects** tab:

- **Migration Data Set Objects in Error.** This zone is only visible if there are objects for this data set in a non-final status that have errors. It indicates the error for each object. A user may use this zone to review errors after the monitor batch job to apply objects completes. Using the error information shown, the user can choose to drill into the record to transition it to **Error Applying** or choose to manually fix the cause of the errors and click **Retry Objects**. The user may also choose to select one or more records to **Reject**.

> **Note**
>
> Refer to Apply Step for more information about resolving errors.

- **Migration Data Set Impacted Object Summary.** This zone is visible once the Import Step has occurred and lists the objects that are related to the data set. To see more information about a specific migration object, click the hypertext for its ID. This brings you to the Migration Object portal. A user may choose to update the status of one or more records by checking the records and clicking **Approve**, **Reject** or **Needs Review** accordingly.

The following zones are visible on the **Errors** tab:

- **Errors Summary.** This zone summarizes migration objects in error by maintenance object, error message category and number. Broadcasting a row provides a more detailed list of distinct error messages.
- **Errors.** This zone lists distinct errors for a selected maintenance object along with a count of migration objects in error with the same exact error message. This information may be used to highlight common configuration issues across objects. A sample migration object is also provided for review. The zone is visible when a selection is made on the summary zone.

The following zones are visible on the **Log** tab, in addition to the standard log entries:

- **Object Retries Summary.** This zone provides a summary of retry processing performed while applying objects. Retrying objects too many times slows down the apply step. Incomplete or invalid export data may result in many objects being retried too many times until they are eventually set to 'Error Applying' status.

## Migration Transaction Portal

This page appears after drilling into a specific migration transaction from the migration data set portal or from the migration object portal.

Refer to Importing and Applying a Migration for an overview of the import process.

The following zones are visible on the main tab:

- **Migration Transaction.** This zone contains display-only information about the selected record. Please see the zone help text for information about this zone's fields.
- **Migration Transaction Objects.** This zone lists the objects that are related to the data set. To see more information about a specific migration object, click the hypertext for its ID. This brings you to the Migration Object portal. A user may choose to update the status of


Copyright © 2026, Oracle and/or its affiliates.

one or more records by checking the records and clicking **Approve**, **Reject** or **Needs Review** accordingly.

- **Entity Summary.** This zone provides a summary count of all migration objects included in the current transaction across the various statuses.

### Migration Object Portal

This page appears after drilling into a specific migration object from the migration data set portal or from the migration transaction portal.

Refer to Importing and Applying a Migration for an overview of the import process.

The **Migration Object** zone contains display-only information about the selected record. Please see the zone help text for information about this zone's fields.

The **Migration Object - List of SQL Statements** zone lists the SQL statements that have been generated as a result of the comparison step. A check mark in the Suppressed checkbox indicates that the SQL for that row will not be applied during the apply step.

The **Migration Object - Algorithms** zone displays any pre-compare algorithms that are associated with the migration plan for this migration object.

### Configuration Deletion

Use the Configuration Deletion portal to review and delete configuration entities that were previously imported and are no longer needed. Refer to Importing and Applying a Migration for an overview of the import process.

Navigate using **Admin**, and then **Implementation Tools**, and then **Configuration Deletion.** You are brought to a portal where you can define a set of import data sets as representing a complete imported configuration and delete entities that were previously imported but not included in the reference set.

The following zones are visible on the **Main** tab:

- **Configuration Summary.** This zone summarizes the number of imported entities by maintenance object that are included in a selected set of import data asets that represent a complete imported configuration. Selecting a maintenance object lists potential entities for deletion.
- **Entities to Delete.** This zone lists potential entities for deletion, i.e. entities of the selected maintenance object that were previously imported but not included in the reference data set. You may select and delete entities from this list.

# Running Batch Jobs

There are several batch jobs that are part of the CMA process, especially the import step (which are highlighted in Import Process Summary). And in some cases, a single batch jobs may process multiple states in the same business object lifecycle. Implementations must decide the best way to manage the batch job submission depending on how they plan to work.

- **Batch scheduler**. If an implementation wishes to put these batch jobs in the batch scheduler, a given job may need to be included several times to manage progressing the records to completion.
- **Timed Batches**. The batch controls can be configured as timed batches so that they run every N minutes based on the setting. This allows for the batch jobs to run periodically and


Copyright © 2026, Oracle and/or its affiliates.

process whatever is ready. A user doesn't have to manually submit a batch request. Navigate to the Batch Control page and select the appropriate batch controls. For each one, change the Batch Control Type to **Timed**. Fill in the additional information that appears for timed batches.

- **Event Driven**. The system provides BO enter plug-in algorithms and batch control post processing plug-in algorithms that automatically submit the appropriate next batch job for that step in the process. This allows for as much automation as possible for the steps that don't require user input. Note that configuration is required because the BOs / batch controls are not configured for this scenario by default. The following table highlights the BO and status where an algorithm may be plugged in and the name of the algorithm to use.

| Business Object | Status | Algorithm |
|---|---|---|
| Migration Data Set Export | **Pending** | `F1-MGDPR-SJ` (Submit Migration Data Set Export Monitor). |
| Migration Data Set Import | **Pending** | `F1-MGDIM-SJ` (Submit Migration Data Set Import Monitor). |
| | **Ready To Compare**, **Retry Objects** | `F1-MGOPR-SJ` (Submit Migration Object Monitor).<br>When using separate batch processes for business data add also this algorithm:<br>`F1-MGOPRB-SJ` (Submit Migration Object Monitor for Business Data). |
| | **Apply Objects** | `F1-MGOAP-SJ` (Submit Migration Object Apply Monitor).<br>When using separate batch processes for business data add also this algorithm:<br>`F1-MGOAPB-SJ` (Submit Migration Object Apply Monitor for Business Data). |
| | **Apply Transactions** | `F1-MGTAP-SJ` (Submit Migration Transaction Apply Monitor). |
| | **Retry Transactions** | `F1-MGTPR-SJ` (Submit Migration Transaction Monitor). |

The following table highlights the batch controls where an algorithm may be plugged in and the name of the algorithm to use.

| Batch Control | Algorithm |
|---|---|
| F1-MGTPR (Migration Transaction Monitor) | `F1-MGDIM-NJ` (Submit Migration Data Set Import Monitor). |
| F1-MGTAP (Migration Transaction Monitor - Apply) | `F1-MGDIM-NJ` (Submit Migration Data Set Import Monitor). |
| F1-MGOAP (Migration Object Monitor - Apply)<br>When using separate batch processes for business data update also this batch control:<br>F1-MGOAB (Migration Object Monitor (Business) - Apply) | `F1-MGDIM-NJ` (Submit Migration Data Set Import Monitor). |


Copyright © 2026, Oracle and/or its affiliates.

| Batch Control | Algorithm |
| --- | --- |
| F1-MGOPR (Migration Object Monitor)<br>When using separate batch processes for business data update also this batch control:<br>F1-MGOPB (Migration Object Monitor (Business)) | `F1-MGTPR-NJ` (Submit Migration Transaction Monitor). |

- **Manual submission**. The user managing the CMA import process submits the appropriate batch jobs on demand when a particular step is ready. Navigate to Batch Job Submission, select the appropriate batch control and fill in the parameters as needed.

Note that for the CMA batch processes used to migrate administrative data, there are multiple important caching considerations. Refer to Caching Considerations for more information.

> ⚠ **Caution**
>
> Be sure that the Thread Count set when submitting the batch job does not exceed the number supported by the thread pool. Otherwise the extra threads will wait until the supported number of threads are finished, possibly resulting in a large number of errors in the Apply steps.

Refer to the parameter descriptions in the batch control metadata for more information about filling in the parameters.

For additional details on submission controls, refer to the topic Batch Job Submission Portal in the Batch Jobs section.

# CMA Reference

This section provides additional reference information.

- Framework-Provided Migration Configuration

## Framework-Provided Migration Configuration

This topic describes special information relating to migration objects provided for use by CMA in the product. Additional objects may be provided by your specific product. Any special information for objects is provided separately in each product's documentation.

The following points highlight some information about Framework-provided migration requests. Navigate to the migration request page in the application to view the details of all provided objects.

- Several base migration requests are supplied to logically group system and administrative tables. For example, there is a migration request for Framework System Configuration **F1-SystemConfig** where most system configuration objects are included. There is another one provided for CMA related configuration objects.
- The system supplies a group migration request `F1-FrameworkConfig` (Framework Configuration), which includes several other migration requests. The expectation is that this migration request includes all the typical objects that are included in a wholesale migration. Your specific product may include this migration request into its own group migration request to support a wholesale migration of all the framework and product administrative tables. An implementation may choose to build a custom group migration


Copyright © 2026, Oracle and/or its affiliates.

request. In this case, review the various migration requests provided by base to see if any may be included as components for the custom migration request. Then any new migration plans added to the base migration request in future releases are automatically included in future migrations.

- There are several different security related migration requests that include different combinations of migration plans to support multiple possible business requirements related to security migration. Note that the security migration request included in the above mentioned group migration request is the one that does not include users. If your implementation wants to copy users to a target environment, refer to Importing Data that References a User for some considerations.

> **Note**
>
> Refer to your specific product's CMA documentation for its recommendation on which migration requests to use for a full migration of framework and product administrative tables.

The following points highlight some information about the Framework-provided migration plans. Navigate to the migration plan page in the application to view the details of all provided objects.

- Fields and characteristic types are not migrated with an object (like a business object or a data area) unless specifically indicated.
- The **Application Service** used by an object is migrated only if it is CM-owned.
- The **Batch Control** object optionally references a User (for 'timed' batches). Refer to Importing Data that References a User for some considerations about copying the user. Also note that when running a batch job, snapshot information is captured on the batch control. Updates like this increment the version number. If a batch control record is part of the migration and the comparison step has detected a change to the batch control, the Apply step will error out for this batch control if a batch job is submitted between the compare and apply step.

> **Note**
>
> CMA batch controls that are part of the import step are executing and as such, the system does not include these records in a migration. If your implementation changes default parameters for any of the batch controls, the recommendation is to manually make those changes to the target region.

- The base migration plans for MO and BO include instructions to copy option types that use foreign key references to refer to other objects. Note that the data stored in the options are not validated, so defining these instructions is not required when doing wholesale migrations. However, including subordinate instructions for foreign key references is useful for targeted migrations to ensure that the related data is included in the migration. If you add additional MO or BO option types that use foreign keys and you want to support targeted migrations, you must create custom migration plans and requests for MO and BO, respectively to include these referenced objects in the migration plan. Note that you do not need to duplicate the instructions in the base migration plans. You may define the additional migration plans to only have the additional custom option types. When submitting a migration request for MO or BO you must include both the base migration plans and the custom migration plans in the request.
- For scripts, schema-based objects and zones, the migration plans provided by the product migrate, through constraints, some of the typical associated data with them. However, data


Copyright © 2026, Oracle and/or its affiliates.

specified through alternate formats (such as through **Edit Data** steps in scripts, referenced in schemas for schema-based objects, or data from mnemonics in zone parameters, etc.) are not identified and combined in the same transaction. The iterative processing functionality of the import step should resolve any timing issues that may result in validation errors for these types of objects.

- There are two migration plans for **Scripts**. The migration plan **F1-ScriptOnly** migrates just the script and its **Application Service** (provided the Application Service is CM-owned). The migration plan **F1-Script** includes most related objects, but does not migrate any objects referenced in the edit data area steps. It does not move the **Function** maintenance object. It may be included in any appropriate custom targeted migration request where scripts and related data should be migrated.
- If your implementation includes a **Feature Configuration** setting for the `F1_DBCONINFO` entry that will be included in a migration request, be sure that the import user on the target region has the appropriate security rights to this entry (**Administrator** access mode for the Feature Configuration application service (**CILTWSDP**).
- The common attachments in the Attachment maintenance object may be considered administrative data to include in a migration. Because this MO has a system generated key, as described in Administrative Data with System Generated Primary Keys, it uses a logical key of the file name and the creation date to determine if the record exists in the target environment. In addition, this MO contains administrative data (common attachments) and non-administrative data (owned attachments). To try to minimize the possibility of key "collision", new common attachments receive a generated key that includes a zero in the middle whereas owned attachments receive a generated key that does not have a zero in the middle.
- The Menu maintenance object has a user defined key, however, its menu lines and menu items have system generated keys. To avoid the possibility of overriding a menu line or menu item incorrectly, the menu MO will check the menu line's menu name in the source and target to be sure they match and will check the menu item's menu line in the source and target to be sure they match otherwise an error will be issued in the comparison step.
- For the system messages, the product provides three different migration plans.
  - Message Category and its Messages (F1-MessageCategory). This migration plan is included in the **F1-SystemConfig** migration request.
  - Message Category (F1-MessageCategoryOnly). This migration plan is provided to support a targeted migration where an implementation has created a custom message category and wants to move it but doesn’t want to move all its messages.
  - Message (F1-Message). This migration plan is provided to support a targeted migration where only specific messages within a message category should be migrated.
- For lookup values, the product provides two different migration plans.
  - Lookup Field and its Values (F1-Lookup). This migration plan is included in the **F1-SystemConfig** migration request.
  - Lookup Value (F1-LookupValue). This migration plan is provided to support a targeted migration where only specific lookup values within a lookup field should be migrated.
- There are some system data objects where no information in a base delivered record may be modified by an implementation. For these records, the base delivered migration requests include selection criteria to only select CM-owned records (because the base records will always exist in the target region assuming both regions have the same release). An example is Algorithm Type. The **F1-SystemConfig** migration request only includes CM-owned algorithm types. However, many system data objects support custom changes to one or more fields, for example the Zone object allows an implementation to


Copyright © 2026, Oracle and/or its affiliates.

override the zone text or certain parameters. Other system data objects support custom additions to a collection. For example, the Maintenance Object allows an implementation to add algorithms or options. For the migration plans related to these system data objects, all records are included in the base delivered migration requests to allow for any customized configuration to be migrated. It means that during the Import / Compare step many base delivered objects that are not customized will be marked **Unchanged**.

- Many of the integration related maintenance objects that include references to environment-specific data, such as Message Senders. This data should be migrated with extreme care. When appropriate, consider taking advantage of URI substitution. Refer to Referencing URIs for more information.

# Facts

Fact is an optional configuration tool for simple workflow-type business messages or tasks. The base package does not provide a dedicated Fact user interface because fact is generic by design. Implementations configure their own user interface to visualize the desired custom business process. The topics in this section describe the generic Fact entity and how it can be customized.

- Fact Is A Generic Entity
- Fact's Business Object Controls Everything
- Fact Supports A Log

## Fact Is A Generic Entity

The Fact maintenance object is a generic entity that can be configured to represent custom entities and support automated workflows for a variety of applications. Each fact references a business object to describe the type of entity it is. A status column on the fact may be used to capture its current state in the processing lifecycle controlled by its business object.

The maintenance object also supports a standard characteristic collection as well as a CLOB element to capture additional information.

## Fact's Business Object Controls Everything

A fact's business object controls its contents, lifecycle and various other business rules:

- Its schema defines where each piece of information resides on the physical Fact maintenance object.
- If may define a lifecycle for all fact instances of this type to follow. Each fact must exist in a valid state as per its business object's lifecycle definition.
- It may define validation and other business rules to control the behavior of facts of this type.

**NOT_SUPPORTED**

For more information about business objects, refer to The Big Picture of Business Objects.


Copyright © 2026, Oracle and/or its affiliates.

## Fact Supports A Log

The Fact maintenance object supports a log. Any significant event related to a Fact may be recorded on its log. The system automatically records a log record when the fact is created and when it transitions into a new state. In addition, any custom process or manual user activity can add log entries.

> **NOT_SUPPORTED**

Refer to State Transitions Are Audited for more information on logging.

# Conversion

This section describes the overall conversion process and the tools provided to support it.

- Understanding The Conversion Process
- Conversion Entity Dashboard

## Understanding The Conversion Process

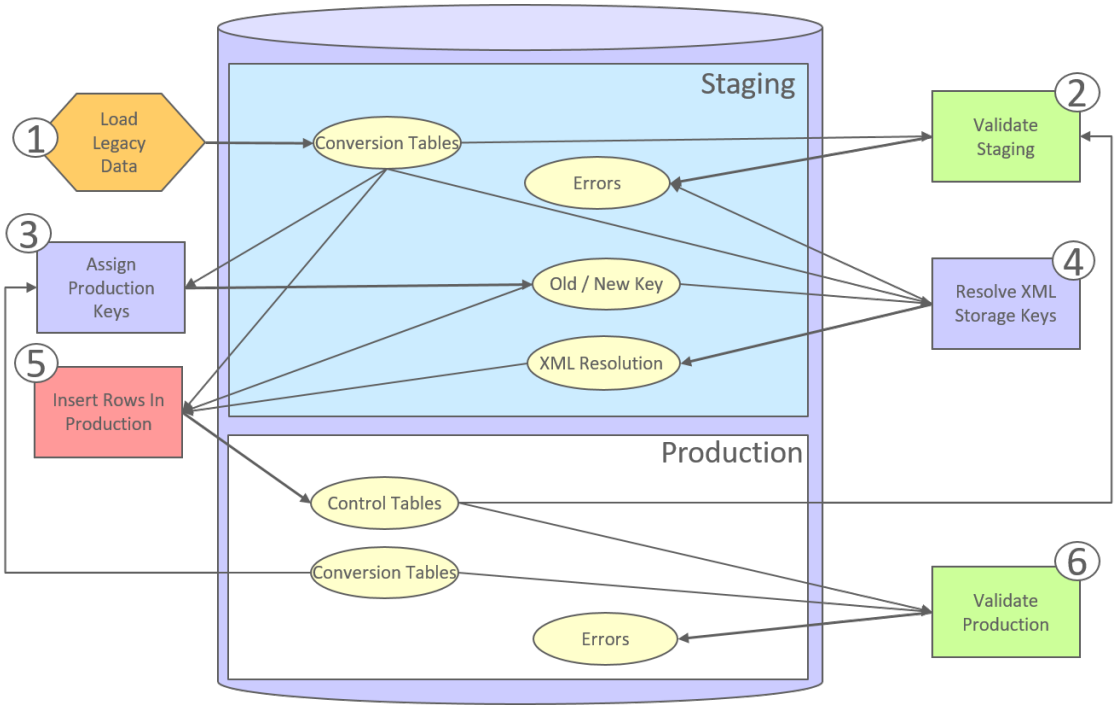
When you're ready to convert data from your legacy system, you will have analyzed your processing requirements according to your business and organizational needs and set up the control tables accordingly. After the control tables are set up, you are ready to load data into the system from your legacy system.

There are two table owners in the system database. We refer to the first owner as **staging** and the second owner as **production**. The staging owner is linked to the tables into which you insert your pre-validated data. The production owner is linked to the tables used by your production system.

> **Note**
>
> It's important to notice that **control tables exist only in production**. In staging, they exist as views to the production schema. Refer to Multiple Owners In A Single Database for more information about table ownership.

The conversion effort involves several steps as illustrated in the following diagram:


Copyright © 2026, Oracle and/or its affiliates.




The following points briefly outline each of the above tasks:

- **Load Legacy Data.** During this step, your legacy data is loaded into the system. Notice that you are not migrating this data directly into production. Rather, your rows are loaded into tables that are identical to the production tables; they just have a different owner. Mapping legacy data into the system is probably the most challenging part of the conversion process because the system is a normalized database (and most legacy applications are not).
- **Validate Staging.** During this step, the system validates the data you loaded into the staging tables. Note that validation rules validate your staging data using the control tables that have been set up in production.
- **Assign Production Keys.** During this step, the system allocates random, clustered keys to the rows in the staging database.
- **XML Resolution.** During this step, the system resolves legacy keys that may be mapped to an XML storage field with new values that were assigned in the key assignment step.
- **Insert Rows Into Production.** During this step, the system populates your production tables with rows from the staging. When the rows are inserted, their prime keys and foreign keys are reassigned using the data populated in the key assignment and XML resolution steps.
- **Validate Production.** During this step, you rerun the object validation processes, but this time against production. We recommend performing this step to confirm that the insertion programs have executed successfully.

> **Note**
>
> Additional steps may be needed to support specific data conversion requirements. Refer to your product documentation for additional conversion related information.

This section further discusses concepts and guidelines related to the conversion process.


Copyright © 2026, Oracle and/or its affiliates.

- Conversion Entities
- Conversion Steps
- A Note About Keys
- Multiple Owners In A Single Database

# Conversion Entities

Conversion is enabled for selected master and transaction maintenance objects. In the same way, only a subset of a maintenance object's tables are enabled for conversion.

The following is provided for each maintenance object that supports conversion:

- An entity validation batch control. This batch process validates data in staging and can also be run to validate data in production.
- A key assignment batch control for each **table** that is eligible for conversion and has a system-generated key.
- An XML resolution batch control if the maintenance object has at least one table that includes an XML storage field.
- An insertion to production batch control for every **table** of the maintenance object that is eligible for conversion.

You may review the conversion configuration provided for an entity and more using the Conversion Entity Dashboard portal.

# Conversion Steps

The following sections provide more details about the steps in the conversion process.

> **Note**
>
> Additional steps may be needed to support specific data conversion requirements. Refer to your product's documentation for additional conversion related information.

- Load Legacy Data Into Staging Tables
- Validate Information In The Staging Tables
- Allocate Production Keys
- XML Resolution
- Insert Production Data
- Validate Production

## Load Legacy Data Into Staging Tables

This section provides some high level discussion about mapping and loading legacy data to the system's staging tables.


Copyright © 2026, Oracle and/or its affiliates.

### Mass Load Utility

Unless in a Cloud installation, you can use any method you prefer to load data from your legacy application. However, we recommend that you investigate your database's mass load utility (as opposed to using insert statements) as the mechanism to load the staging tables. In addition, we strongly recommend that you disable the indexes on these tables before populating these tables and then enable the indexes after populating these tables.

> **Note**
>
> In a Cloud installation, refer to **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

### Populating Keys

Some tables in the staging database have keys that are system-assigned random numbers. Special consideration should be given when populating these keys. Refer to A Note About Keys for more information on how to properly populate these tables. Those tables that don't have system-assigned random numbers have keys that are a concatenation of the parent's prime-key plus one or more additional fields.

### Populating Characteristic Tables

There are many maintenance objects that include a characteristic table used to capture miscellaneous information about the object. Most of these tables include an indexed column used when searching by characteristic value called Search Characteristic Value. During conversion and depending on the type of characteristic, this column must be populated as follows:

- **Predefined.** Populate search characteristic value with the contents of the characteristic value column converted to upper case.
- **Ad hoc.** Populate search characteristic value with the first 50 bytes of the ad hoc characteristic value column converted to upper case.
- **Foreign key.** Populate search characteristic value by concatenating the values of each foreign key characteristic value column to a maximum of 50 bytes.

## Validate Information In The Staging Tables

During the first validation step, the system validates the data you loaded into the staging tables. This section provides some high level discussion about the validation process.

### A Batch Process Per Maintenance Object

Each of the maintenance objects that are eligible for conversion must be validated using its respective entity validation conversion batch process. These are classic background processes that can also be run against production data.

These processes are multi-threaded and have no dependency on other processes.

### Validation Mode

By default, each entity is fully validated for schema as well as entity validation rules, at the maintenance object and business object level. The batch process performs the same business


Copyright © 2026, Oracle and/or its affiliates.

rules that are used to validate data added by users in your production system. You may restrict the batch process to perform schema validation only or entity validation rules only using a batch parameter. This may be useful in identifying common data issues in preliminary runs.

The batch process sets an indication available to validation rules, that they are being called as part of a conversion process. A custom validation algorithm may use this indication to skip unnecessary rules that are not applicable as part of conversion.

> **Note**
>
> The indication is accessible via the scripting $$isConvGenericValidation context variable.

#### Random Sample Mode

By default all records are selected for validation. The "Process Every Nth" batch parameter allows you execute this process in a random-sample mode to highlight pervasive errors. In this mode, you are actually telling the batch process to validate every N records.

We strongly recommend validating each entity in the following steps:

- Execute the validation batch process in random-sample mode.
- Review and correct the errors. Note, you can use the base package's transactions to correct an error if the error isn't so egregious that it prevents the object from being displayed on the browser.
- After all pervasive errors have been corrected; re-execute each object's validation batch process in all-instances mode to highlight elusive, one-off errors.

#### Reported Errors

Errors encountered during validation are logged onto the conversion Validation Error (CI_VAL_ERR) table. Note that at the start of this job, all rows in the conversion error table for the process maintenance object are deleted.

You can view errors highlighted by the validation process using the Conversion Entity Dashboard portal.

#### Uneven Thread Distribution Using Legacy Keys

By default, the 'Object Validation' batch process distributes thread processing load using the **Thread Level SQL Select** (THDS) batch strategy. This method relies on the entity's prime key to be evenly distributed.

In the staging schema, entities still reference their legacy prime keys which may not be evenly distributed. When the distribution skew is significant, the performance of this batch process may be impacted. In this situation, you may want to use the **Key Ranges** (KEYS) batch strategy instead. Refer to Parallel Background Processesfor more information.

To do so, add the **Batch Strategy** common batch parameter to the corresponding 'Object Validation' batch control record and set it to the 'KEYS' value.

#### Recommendations To Speed Up Validation

The following points describe ways to accelerate the execution of the validation process:

- Ensure that statistics are recalculated after data has been inserted into the staging tables.


Copyright © 2026, Oracle and/or its affiliates.

- Execute the process multi threaded.
- Execute shorter running validation processes (e.g., less records) first so that the error data can be analyzed while other processes are busy running.
- Remember that validation can be run in random-sample mode. We recommend running these batch processes using a large sample value for this parameter until the pervasive problems have been rectified.

> **Note**
>
> In a Cloud installation, refer to **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

#### Another use for these programs

In addition to validating your objects after conversion or an upgrade, the validation programs have another use. Say for example, you want to experiment with changing the validation of a business entity and you want to determine the impact of this new validation on your existing records. You could change the validation and then run that entity's validation batch process- it will produce errors for each record that fails the new validation.

## Allocate Production Keys

During the key assignment step, the system assigns a new key for each legacy key for a table that its prime-key requires a system-generated random key. The conversion process allocates new prime keys to take advantage of the system's parallel processing and data-clustering techniques in the production system (these techniques are dependent on randomly assigned, clustered keys).

The topics in this section provides some high level discussion about the key assignment process and describe the background processes used to assign production keys to the staging data.

#### The Old Key / New Key Table

It's important to understand that the system does not overwrite the prime-keys on the rows in the staging database, as this is a very expensive IO transaction. Rather, a series of tables exist that hold each row's old key and the new key that will be assigned to it when the row is transferred into the production database. We refer to these tables as the "old key / new key" tables.

The convention "<1st letter of owner flag>K_<table_name>" is used to denote the old key / new key table name. For example, the old key / new key table for CI_ACCT is called CK_ACCT.

The insertion batch process that transfers the rows into the production database use the new key for the main record of the key along with any other record where this key is a foreign key. In the same way, the XML resolution process resolves conversion foreign keys residing in XML storage fields and replaces them with their corresponding new keys from these table.

#### A Batch Process Per Table

A key assignment batch process is provided for each table that has a system-generated key and belongs to a maintenance objects that is eligible for conversion. The batch process is responsible for populating the corresponding old key / new key table (in other words, you don't


Copyright © 2026, Oracle and/or its affiliates.

have to populate these tables). These processes are single threaded. Refer to a table's corresponding key assignment batch process for more information.

### Key Assignment Dependencies

Most tables with system-generated keys do not inherit part of their key from another table's key. Their corresponding key assignment batch process have no dependencies and can therefore be executed in any order you please.

Some tables inherit part of their key from another table's key. Key assignment batch processes for such tables must be executed in key inheritance order. In other words, the key assignment process for a table should be run after the process that generates keys for the entity it depend on. The Conversion Entity Dashboard portal displays key inheritance dependency information if any exist for a conversion entity.

> **Note**
>
> You may run multiple key assignment batch process in parallel as long as they are independent with respect to key inheritance.

### Iterative Conversions

Rather than perform a "big bang" conversion (one where all records of an entity are populated at once), some implementations have the opportunity to go live on subsets of their entity base. If this describes your implementation, please be aware that the system takes into account the existing prime keys in the production database before it allocates a new key value. This means when you convert the next subset of customers, you can be assured of getting clean and unique keys.

Key assignment logic creates the initial values of keys by a manipulation of the sequential row number, starting from 1. After any conversion run, a subsequent conversion run will start with that row number again at 1, and the possibility of duplicate keys being assigned will be higher. The key assignment batch process allows you to specify a starting value for that row number. The purpose of this parameter is to increase the value of row number by the given value, and minimize the chance of duplicate key assignment. This parameter is only used if you are performing conversions where data already exists in the tables in the production database.

### Run Type

Key assignment is performed in two steps:

- **Initial Key Generation.** The system allocates new keys to the rows in the staging tables (i.e., it populates the respective old key / new key table).
- **Duplicate Key Resolution.** The system reassigns keys that are duplicates compared to production.

By default, both steps are performed at the same run but you have the option to run them separately by indicating which step to run via a batch parameter. The proper use of this parameter will greatly speed up the key assignment step as described under **Recommendations To Speed Up Key Generation** section.

### Recommendations To Speed Up Key Generation

The following points describe ways to accelerate the execution of the key generation programs.


Copyright © 2026, Oracle and/or its affiliates.

- For a non-Cloud installation:
  - Make the size of your rollback segments large. The exact size is dependent on the number of rows involved in your conversion. Our research has shown that processing 7 million rows generates roughly 3GB of rollback information.
    * Setup the rollback segment(s) to about 10GB with auto extend to a maximum size of 20GB to determine the high water mark
    * A next extent value on the order of 100M should be used.
  - Make sure to turn off all small rollback segments (otherwise Oracle will use them rather than the large rollback segments described above).
  - After the key assignment programs execute, you can reclaim the space by:
    * Keep a low value for the "minimum extent" parameter for the rollback.
    * Shrink the rollback segments and the underlying data files at the end of the large batch jobs.
- Compute statistics on the old key / new key tables after every 50% increase in table size. Key generation is performed in tiers or steps because of the inheritance dependency between some tables and their keys. Although key generation for the inheritance dependency tier currently being processed is performed by means of set-based SQL, computation of statistics between tiers will allow the database to compute the optimum access path to the keys being inherited from the previous tier's generation run.
- Optimal use of the **Run Type** batch parameter.
  - Before any key assignments, alter both the "old key" F1_CX_ID index and the "new key" CI_ID index on the old key / new key tables to be unusable.
  - Run all key assignment batch processes in tiers, submitting each job to only perform the **Initial Key Generation** step.
  - Rebuild the indexes on the old key / new key tables. Rebuilding the indexes using both the PARALLEL and NOLOGGING parameters will speed the index creation process. Statistics should be computed for these indexes.
  - Run all key assignment batch processes in the current tier that were previously run in initial key generation mode, to perform the **Duplicate Key Resolution** step. This will reassign all duplicate keys.

> **Note**
>
> In a Cloud installation, refer to **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

## XML Resolution

Most system-generated foreign keys are stored in physical fields and characteristic tables and as such their legacy value is replaced with its corresponding new key as part of the insertion to production process. Rarely, maintenance objects may store system-generated foreign keys in an XML storage field, in other words, a field defined with the data type of CLOB or XML. This step is only applicable to such maintenance objects.


Copyright © 2026, Oracle and/or its affiliates.

During this step, the system resolves convertible system-generated foreign keys that may reside in XML storage fields you may have loaded into the staging tables. This section provides some high level discussion about the XML resolution process.

### The XML Resolution Table

It's important to understand that the system does not overwrite the prime-keys on the XML storage fields in the staging database, as this is a very expensive IO transaction. Rather, a corresponding XML resolution table exists for each table that defines an XML resolution field to capture each row's resolved XML storage content, in other words, the content with all the old keys replaced with the new assigned keys.

The convention "<1st letter of owner flag>R_<table_name>" is used to denote the XML resolution table name.

The insertion batch process that transfers the rows into the production database replaces each XML storage field with its resolved value from the corresponding XML resolution table.

### A Batch Process Per Maintenance Object

Each of the maintenance objects that are eligible for conversion and support XML storage fields is provided with an XML resolution batch process. These batch processes must be run to resolves foreign keys that may reside in these XML storage fields.

These processes are multi-threaded and must be executed after the key assignment step has completed and before inserting data to production.

### XML Resolution Eligibility

Not all maintenance objects that support XML storage fields actually store convertible system-generated foreign keys in their XML storage field. If none of the business objects associated with the maintenance object involve mapping of such foreign keys to an XML storage field then XML resolution is not needed for any row in the maintenance object. The XML resolution batch process detects such situation and completes right away without storing any rows in any of the maintenance object's XML resolution tables.

> **Note**
>
> If the XML resolution batch process detects that the maintenance object is eligible, the maintenance object must reference a physical BO. If a product delivered business object supports system-generated foreign keys in an XML storage field, the product should be supplying a physical BO. If your implementation has system-generated foreign keys in an XML storage field in a custom business object for a maintenance object where the product has not supplied a physical BO, you must create a physical BO and link it to the maintenance object.

### Only Resolved Values Are Captured

XML storage fields typically store large amounts of data. To avoid capturing the same XML content redundantly, the system only stores values in the resolved XML storage fields if the resolved value is different than the original value, in other words, if at least one key was resolved.

If for a given record the resolved XML content is the same as the original content then the following rules apply:

- If this is the primary table of the maintenance object a record is inserted into its corresponding XML resolution table for that record with no value in the XML storage field.


Copyright © 2026, Oracle and/or its affiliates.

- If this is a child table of the maintenance object then no record is inserted into the corresponding XML resolution for that record.

#### Reported Errors

Errors encountered during XML resolution are logged onto the conversion Validation Error (CI_VAL_ERR) table. Note that at the start of this job, all rows in the conversion error table for the process maintenance object are deleted.

You can view errors highlighted by the XML resolution process using the **Validation Error Summary** page.

#### Uneven Thread Distribution Using Legacy Keys

By default, the 'XML resolution' batch process distributes thread processing load using the **Thread Level SQL Select** (THDS) batch strategy. This method relies on the entity's prime key to be evenly distributed.

In the staging schema, entities still reference their legacy prime keys which may not be evenly distributed. When the distribution skew is significant, the performance of this batch process may be impacted. In this situation, you may want to use the **Key Ranges** (KEYS) batch strategy instead. Refer to Parallel Background Processesfor more information.

To do so, add the **Batch Strategy** common batch parameter to the corresponding ' XML resolution' batch control record and set it to the 'KEYS' value.

## Insert Production Data

The topics in this section describe the background processes used to populate the production database with the information in the staging database.

#### A Batch Process Per Table

An insert to production batch process is provided for each table of a maintenance objects that is eligible for conversion.

The batch process is responsible for transferring to production all the rows from the staging table while replacing all references to old keys with their corresponding new keys as follows:

- Keys residing in prime keys, foreign key fields and characteristics are resolved using the corresponding old key / new key tables.

  > **Important**
  >
  > If metadata constraints are missing, then FK references for physical fields may be missed. It is important to review each conversion table that it has all the necessary constraints defined.

- Keys residing in generic foreign key reference tables where each row represents a reference to an entity's maintenance object and prime key values are also resolved using the corresponding old key / new key tables. Note that all entities referenced this way must either be convertible or view of production. A reference to an entity that is explicitly marked as not convertible is not supported as it cannot be resolved.
- Keys residing in XML storage fields are resolved using the corresponding XML resolution tables.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Foreign keys residing in any other form of storage are not resolved. For example, any references stored in raw elements, log records, or any other field that is not used as an XML storage field.

All insertion batch processes are independent and may run concurrently. Also note, all insertion batch processes can be run in many parallel threads as described in the next section (in order to speed execution).

**Recommendations To Speed Up Insertion**

The following points describe ways to accelerate the execution of the insertion batch processes:

- Before running the first insertion batch process:
  - Rebuild the index on the prime key on the old key / new key table.
  - Re-analyze the statistics on the old key / new key table.
  - Alter all indexes on the production tables being inserted into to be unusable.
- After the insertion programs have populated production data, rebuild the indexes and compute statistics for these tables.

> **Note**
>
> In a Cloud installation, refer to the **Data Conversion Support for Cloud Implementations** for more information about the various tools provided to support database related conversion tasks.

## Validate Production

During this step, you rerun the object validation batch processes, but this time in production. We recommend rerunning these batch processes to confirm that the insertion batch processes have executed successfully. We recommend running these batches in random sample mode (e.g., validate every 1000th object) rather than conducting a full validation in order to save time. However, if you have time, you should run these processes in full validation mode (to validate every object).

# A Note About Keys

The prime keys of the tables in the staging database are either system-assigned random numbers or they aren't. Those tables that don't have system-assigned random numbers have keys that are a concatenation of the parent's prime-key plus one or more additional fields. Every table whose prime key is a system-assigned random number has a related table that manages its keys; we refer to these secondary tables as "key tables".

The following points provide more information about the key tables:

- Key tables are used by programs that allocate new keys. For example, before a new key is allocated, the key assignment program checks the corresponding key table to see if it exists.


Copyright © 2026, Oracle and/or its affiliates.

- Key tables only have two columns:
  - The key of the object.
  - An environment ID. The environment ID identifies the database in which the object resides.
- Key tables are named the same as their primary table with a suffix of "_K". For example: The key table for CI_ACCT is CI_ACCT_K.
- The key table for a table whose prime key is a system-assigned is defined on its table definition record.
- When you populate rows in tables with system-assigned keys, you must also populate a row in the related key table. For example, if you insert a row into CI_ACCT, you must also insert a row into CI_ACCT_K. The environment ID of these rows must be the same as the environment ID on this database's installation record.
- When you populate rows in tables that reference this record as a foreign key, you must use the appropriate key to ensure the proper data relationships. For example, if you insert a row in CI_SA for the above account, the ACCT_ID column must contain the temporary account key.
- When you insert rows into your staging database, the keys do not have to be random, system-assigned numbers. They just have to be unique. A later process, Allocate Production Keys, will allocate random, system-assigned keys prior to production being populated.

# Multiple Owners In A Single Database

The conversion process relies on the following table owner configuration in the system database:

- The production owner is linked to the tables used by your production system. These tables have an owner ID of CISADM.
- The staging owner is linked to the tables into which you insert your pre-validated data. These tables have an owner ID of CISSTG.

The staging owner schema is almost identical to the production schema, with the following exceptions:

- Control tables are not actual tables in staging but rather views to the corresponding tables in production.
- Conversion specific tables designed solely to support the conversion process exist only in the staging schema. For example, tables that manage key assignment and XML resolution.

This section provides some high level concepts related to these table owners.

**Validation Always Uses Production Control Data**

When the validation batch processes run against your staging data, they validate the staging data against the production control tables (and insert errors into the staging error table). This means that the SQL statements that access / update entities needs to use the staging owner whereas the SQL statements that access control tables need to use the production owner. But notice that when these same validation batch processes run against production, the SQL statements will never access the staging owner. Rather, they all point at the production owner.

This is accomplished as follows:


Copyright © 2026, Oracle and/or its affiliates.

- A separate application server must exist for each owner. Each application server points at a specific database user ID.

> **Note**
>
> In a Cloud installation, the application server may only point to a specific database user ID at any point in time. Refer to **Data Conversion Support for Cloud Implementations** for more information about switching owners.

- The database user ID associated with the staging owner uses CISSTG as the owner for the master and transaction tables, but it uses CISADM as the owner of the production control tables.
- The database user ID associated with the production owner uses CISADM as the owner for all tables.

You may wonder why we went to this trouble. There are several reasons:

- We wanted to reuse the validation logic that exists in the programs that validate your production data. In order to achieve this, these programs must sometimes point at the staging owner, and other times they must point at the production owner (and this must be transparent to the programs otherwise two sets of SQL would be necessary).
- We wanted to let you use the application to look at and correct staging data. This can be accomplished by creating an application server that points at your staging database with the ownership characteristics described above.
- We wanted the validation programs to be able to validate your production data (in addition to your staging data). Why would you want to validate production data if only clean data can be added to production? Consider the following scenarios:
  - After an upgrade, you might want to validate your production data to ensure your pre-existing user-exit logic still works.
  - You may want to conduct an experiment of the ramifications of changing your validation logic. To do this, you could make a temporary change to user exit logic (in production) and then run the validation jobs on a random sample basis.
  - You forget to run a validation program before populating production and you want to see the damage. If you follow the instructions in this document, this should never happen. However, accidents happen. And if they do, at least there's a way to determine the ramifications.

#### Only Validation Can Work In Both Owners

While the redirection of owner ID's is a useful technique for the validation batch processes, it cannot be used by the key assignment and production insert batch processes? Why, because these processes have to access the same tables but with different owners at the same time. They also need to reference conversion specific tables that do not exist in production. For example, the batch process that inserts rows into a table in production must select rows from the staging version of that table, resolve keys from the conversion old key / new key mapping tables and insert the resolved records into the production version of that same table

This is accomplished as follows:

- In staging, a view to production exists for each eligible to conversion table. These views have hard-coded the database owner to point to production. For example, there is a view called CX_PER that points at person table in production.


Copyright © 2026, Oracle and/or its affiliates.

- The key assignment and insertion programs use these views whenever then need to access production data.

## Conversion Entity Dashboard

This portal provides an overview of a maintenance object's conversion configuration along with information related to its current status and progress within the overall conversion process.

Refer to Understanding Conversion for more information.

You can access the portal from the **Admin Menu**, and then **Implementation Tools**, and then **Conversion Entity Query.** You are brought to a query portal with options for searching for a specific maintenance object that is eligible for conversion. Once a maintenance object has been selected you are brought to the conversion entity dashboard portal to review the information for the selected record.

The following zones may appear as part of the portal's **Main** tab page.

- **Tables.** This zone provides information about tables that belong to the maintenance object and are marked as eligible for conversion.
- **Batch Controls.** This zone lists the conversion batch controls defined for the maintenance object in the order they should be executed within the conversion process along with current execution details if any.
- **Error Summary.** This zone provides a summary of all conversion errors related to the maintenance object.
- **Duplicate Keys.** This zone lists duplicate legacy keys that need to be resolved by the key assignment conversion process.

The **Errors** tab page provides a query zone that allows you to review conversion errors related to the maintenance object by various criteria.

# Data Export

This section describes concepts and tools related to the various types of data export methods supported by the product.

- Understanding Data Export
- Defining Data Export Control
- Generalized Export Dashboard

## Understanding Data Export

The application supports the following classes of export methods for a maintenance object:

- **Generalized.** This method may be used for maintenance objects that are low to medium in volume and update activity. Most maintenance objects are eligible for this method.
- **Specialized.** This method should be used for maintenance objects that are high in volume or very frequently updated.


Copyright © 2026, Oracle and/or its affiliates.

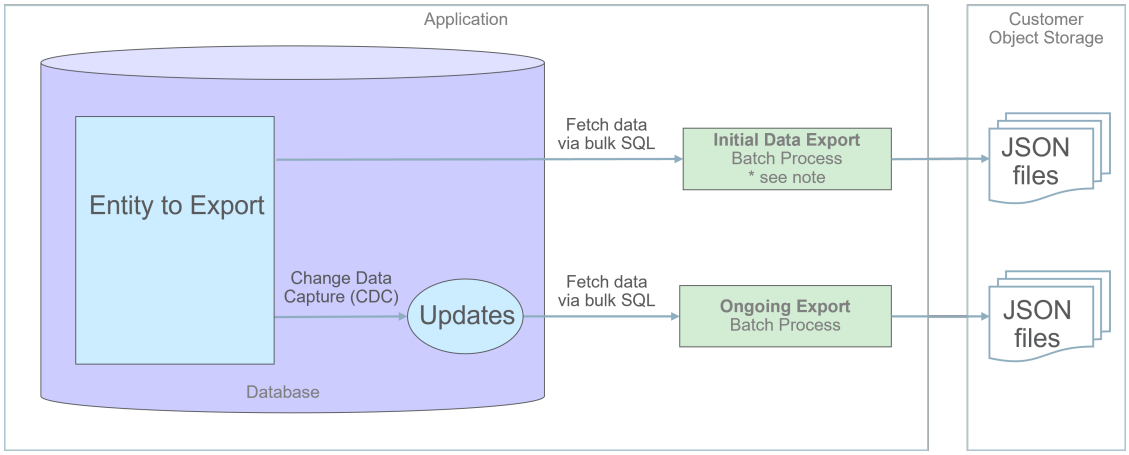
> **Note**
>
> It is your organization's responsibility for designing the process by which these files are imported to one or more target environments.

This section discusses concepts and guidelines related to data export features.

- Generalized Data Export
- Specialized Data Export

# Generalized Data Export

The generalized data export method provides a file-based export of data in JSON format. Exporting the data of a maintenance object using this method involves an initial export of the entity's entire data followed by an ongoing export of changes as they are made over time. These processes are illustrated in the following diagram:




The overall process involves the following steps:

- Enable entities for export in the Generalized Export Dashboard portal. You may enable many entities at once. This step also allows you to enable ongoing export of these entities.
- Submit the Generalized Initial Export Initiator (`F1-GEXPI` ) batch process to export the current content of all enabled entities. This step submits a separate batch process for each entity you have enabled for export.

  > **Note**
  >
  > A full database export can be used instead. The export file can be imported in the target environment to establish the initial baseline. Cloud customers may contact cloud operations to request such export for on-premise use. This schema export is limited for reporting purposes and is intended for initial extract only.

- Schedule the Generalized Ongoing Export (`F1-GEEXO` ) batch process to run periodically.

The following sections further discusses concepts and guidelines related to the generalized data export method.


Copyright © 2026, Oracle and/or its affiliates.

### Eligibility

While most entities are eligible for generalized data export, not all of them are. Here are a few of the common reasons for that:

- The entity manages extremely high volume of data. Tracking changes to such entity would double the volume of managed data and impact overall performance.
- The entity is updated very frequently. Tracking these events may impact overall performance.
- The entity is used by the product's infrastructure and operational processes. These entities are maintained differently and cannot be subject to the generalized data export methods.

By default a maintenance object is eligible for the generalized data export method unless explicitly marked otherwise by the **Data Export Class** option. The option may be used to mark an entity as either not allowed for any type of export or allowed for a specialized export only. The lack of such explicit option means the maintenance object is eligible for the generalized export.

> ⚠ **Caution**
>
> If an entity is explicitly marked as not eligible for generalized data export there is a good reason for that and its configuration should not be changed.

### One Method Fits All

The generalized data export method is designed to accommodate many eligible maintenance objects in a generic way that must also takes performance considerations into account. For this reason this method does not support entity specific filtering options nor custom rules of any kind.

The following rules apply to all entities that may be exported using this method:

- Entire data is exported.
  - All fields on all tables that belong to the maintenance object are exported except for "key" and "log" tables if any.
  - Maintenance object log tables typically do not contain useful business information and as such are omitted from export for performance and data volume reasons. However, there are entities for which log records provide analytical value. When applicable, use the **Export Log Tables** option to explicitly include log tables for a specific maintenance object.
- Same data structure and format is used for all entities. Refer to the **Record Format** section for more information.
- Data is exported to files only. Refer to the **Export To Files Only** section for more information.
- All changes are captured. It is not possible to exclude some changes from being tracked for ongoing export based on custom business rules.
- When data changes, a full snapshot of the data related to that entity is exported to avoid heavy data merging on the target side.
- When the entity is deleted, only the primary key of the entity is exported along with an indication that it is deleted.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Export is not supported for entities that store fields in a time zone other than the base time zone. This type of definition is not commonly used and includes maintenance objects that at least one of their fields is defined as stored in **Reference Time Zone**.

### Export To Files Only

The immediate destination of both initial and ongoing export processes are files in a customer specified location. In a Cloud installation files are created in a customer owned Object Storage.

Based on your business requirements you may further consume these files by downstream applications such as data lakes etc.

### Enabling Data Export

To enable data export for an eligible maintenance object a **Data Export Control** record needs to be created. The record keeps an indication of whether the initial export for the maintenance object has been completed or not and whether changes made to the entity should be tracked and exported on an ongoing basis. A designated portal allows you to enable the generalized data export for many maintenance objects at once as well as monitor their current export status at a glance. Refer to Generalized Export Dashboard portal for more information.

> **Caution**
>
> Enabling and disabling entities for export should only be made via the Generalized Export Dashboard or the Data Export Control portals to ensure the configuration is valid and complete.

> **Caution**
>
> When applicable, ongoing data export should be enabled before submitting the initial export batch process for the entity. This ensures that changes made during the initial export are properly tracked.

Ongoing export options are cached in various online and batch caches. In order for these options to take effect and track entity changes made by any online or batch process, the various caches must be properly flushed when these options are changed. Using the designated portals to enable entities for export triggers a 'global' flush of the caches. If the batch thread pool workers are configured to refresh their caches when a global flush is requested, then this is the only step required. If not, then the `F1-FLUSH` batch job should also be submitted to refresh the caches used in batch processing. Refer to Caching Overview for more information.

### Initial Data Export

Each maintenance object that is eligible for the generalized data export method is associated with an initial data export batch control that is referenced on the maintenance object using the **Export Batch Control** option. The same common batch program is used by all these batch controls to export the entire data of an entity to files. The process is multi-threaded and produces one file per thread by default. An optional batch parameter allows for multiple smaller files to be generated per thread by setting a limit on the number of records written to each file. Refer to the **File Size** section for more information.


Copyright © 2026, Oracle and/or its affiliates.

Instead of manually submitting the generalized initial export batch process for each maintenance object you have enabled for export, submit the F1-GEXPI Generalized Initial Export Initiator batch process to automatically submit them all at once.

The initial export process for each maintenance object updates the corresponding data export control record for the maintenance object to indicate that the process has started and updates it again upon completion. This allows you to view the overall status of the initial export across all enabled entities on the Generalized Export Dashboard portal.

Note that the initiator batch process only submits batch processes for enabled maintenance objects that according to their data export control record their initial export has not started yet. If you need to export one or more entities again, reset the initial export status indication on their data export control records and submit the initiator batch process again.

The initial export file name is constructed as follows:

```
INIT_EXPORT[_file prefix(optional)]_[maintenance object]_[batch
run]_[thread]_[thread count]_[timestamp][_file sequence].json.gz
```

The "file sequence" part of the file name is used only when more than one file is created per thread. The sequence suffix is populated starting from the second file as follows _2.

Refer to any of the individual generalized initial export batch controls and the initiator batch control for more information.

#### Restricting Initial Export By Date

By default, the Initial Data Export batch process exports the entire data for an entity. In some situations, typically around high volume historical data, there may be a need to restrict the export to more recent data, for example the last few months of data.

The initial export batch process supports an optional batch parameter **Restrict By Date** that can be used to limit the scope of data to export. The parameter references a date field and a requested time frame in terms of number of days prior to business date. Refer to the F1-GEIXP Generalized Initial Export Template batch control for more information.

> ⚠ **Caution**
>
> Excluding records from export should be carefully considered as this may result in data discrepancies when the excluded data is referenced on other exported entities.

> ⓘ **Note**
>
> The option is only applicable to specific entities that have a qualifying reference date and the time frame for export may vary for each customer. Therefore, the new parameter is not added to any base product Initial Export batch control aside of the template batch control for documentation purposes. You may clone the base product batch control for a specific entity and add the parameter as per your business needs.

#### Ongoing Data Export

Data changes are tracked by the Capture Change for Ongoing Export (F1-MO-REGCHN) Maintenance Object Audit algorithm. The algorithm records the prime key of a changed entity


Copyright © 2026, Oracle and/or its affiliates.

in a designated **Data Export Update** table that serves as a queue of changed entities to export. The queue is later on consumed by the ongoing data export batch process.

> **Note**
>
> The Maintenance Object Audit plug-in spot is capable of tracking changes made to the entity by any layer of the application. Making changes using direct SQL would bypass this mechanism and therefore should be avoided.

The audit rule is added to a maintenance object when the ongoing data export option on its corresponding data export control record is enabled. In the same way the audit rule is removed from the maintenance object when the ongoing export option is disabled.

> **Note**
>
> The ongoing export option has to be enabled on the data export control record for the entity before it is initially exported. This ensures changes made while the initial export process is in progress are not missed.

Unlike the initial export batch process that requires a batch control to be defined for each eligible maintenance object, a single batch process is used to export ongoing changes made to all entities.

The following describes the ongoing export batch process at a high level:

- All changes queued in the Data Export Update table are exported to a designated ongoing export file. Refer to the **Managing a Large Backlog** section for more information.
- A full snapshot of the data related to the changed entity is exported to avoid heavy data merging on the target side.
- Once exported, records are deleted from the Data Export Update table.

The process is multi-threaded and a separate file is produced for each thread by default. An optional batch parameter allows for multiple smaller files to be generated per thread by setting a limit on the number of records written to each file. Refer to the **File Size** section for more information.

> **Note**
>
> It is important to schedule the ongoing export batch process to run frequently enough to avoid accumulating a high volume of records in the Data Export Update table. The smaller the table is kept the faster it is for the change data capture mechanism to add records to it.

The ongoing export file name is constructed as follows:

```
INC_EXPORT_[batch run]_[thread]_[thread count]_[timestamp][_file
sequence].json.gz
```

The "file sequence" part of the file name is used only when more than one file is created per thread. The sequence suffix is populated starting from the second file as follows _2.


Copyright © 2026, Oracle and/or its affiliates.

Refer to Generalized Ongoing Export (F1-GEEX0) batch control for more information.

### File Size

The initial export batch process produces a single file per thread by default. For some high volume entities this may result in files that are too large to manage. To produce multiple and smaller files for a specific entity, limit the number of records written to each file by setting the corresponding parameter on the initial export batch control defined for the entity. The same parameter may be provided when submitting the Initial Export Initiator batch process, in which case the initiator value is used only when a corresponding value has not been defined on the entity's specific batch control record.

In the same way, the ongoing export batch process produces a single file per thread by default. To produce multiple and smaller files, limit the number of records written to each file by setting the corresponding parameter on the batch control.

To avoid the creation of too many files by a single batch run, the system sets a maximum limit of around 500 files per run. When the requested number of records per file is determined to cause the number of generated files to exceed the system limit, the actual value used is adjusted to meet the system limit.

### Managing a Large Backlog

When a large backlog of changes accumulates in the ongoing export queue for some reason, the next batch takes longer to export which delays the downstream import step. A better option is for the backlog to be cleared in smaller "chunks", allowing the downstream process to import them as they are made available.

You may limit the number of records the batch process exports using the **Thread Processing Limit** batch parameter. When specified, the number of records exported by each thread is capped by the specified limit, leaving the remaining records to be consumed by subsequent batch runs. Instead of waiting for the next scheduled batch run, a new batch process may be automatically submitted, if explicitly requested to do so using the **Automatic Backlog Clearance** batch parameter.

It is important to ensure that the backlog clears gradually and that the queue resumes its normal size in a timely manner. The processing limit must therefore be set with an appropriate value that would promote proper clearance of the queue. To do so, at least 20% of the queue, and no less than 100,000 records, must be consumed by each batch run across all threads. If the specified limit does not meet the minimum clearance requirement then the processing limit is adjusted accordingly.

### Record Format

Data is fetched and exported directly from the database in bulk read operations for performance reasons. For that reason, the structure used does not reflect the logical data model of the entity but rather its physical list of tables. Data for a given instance is organized by tables followed by rows in each table and for each row its list of fields in JSON format.

The following describes the format and structure used:

```
{
"OBJ": "<mo name>",
"TIMESTAMP": "<export time in ISO format for example
2019-07-25T11:06:04.740615Z>",
"PK1": "<mo pk1 value>",
"PK2": "<mo pk2 value if any>",  ← PK2-5 should only be included when
applicable
"PK3": "<mo pk3 value if any>",
```


Copyright © 2026, Oracle and/or its affiliates.

```
"PK4": "<mo pk4 value if any>",
"PK5": "<mo pk5 value if any>",
"DELETED": true,  ← should only be included when the entity is deleted
"DATA":
{
"<MO table name 1>":
[
{<name value pairs of all fields in row 1 in that table>},
{<name value pairs of all fields in row 2 in that table>},...
{<name value pairs of all fields in row n in that table>},...
],
"<MO table name 2>":
[
{<name value pairs of all fields in row 1 in that table>},
{<name value pairs of all fields in row 2 in that table>},...
{<name value pairs of all fields in row n in that table>}
],...
 "<MO table name n>":
[
 {<name value pairs of all fields in row 1 in that table>},
{<name value pairs of all fields in row 2 in that table>},...
{<name value pairs of all fields in row n in that table>}
]
}
```

Notes about field values and formats:

- All fields for a row are included even when the field is empty or null.
- All string values are trimmed of excess trailing spaces but if a field is empty it exported as a single space and not null.
- An empty array is used for a table node when no records exists for the entity in the table.
- Date/time information is converted to UTC time zone and exported in ISO format for example 2019-07-25T11:06:04.740615Z.
- Deletes have an explicit indicator in the header.

### Order of Consuming Files

It is important to apply all the initial export files for a maintenance object before applying ongoing change files to ensure data consistency.

The following is a recommended procedure for consuming generalized data export files:

- Apply all initial data export files before ongoing export files.
- Apply initial files for a maintenance object in run number order.
- Apply ongoing export files in run number order.
- If a new initial export is available the file consumption procedure should be changed temporarily as follows:
  - Stop consumption of ongoing export files.
  - Apply the new initial export files.
  - Resume consumption of ongoing export files.


Copyright © 2026, Oracle and/or its affiliates.

### Recover from a Lost Ongoing Export File

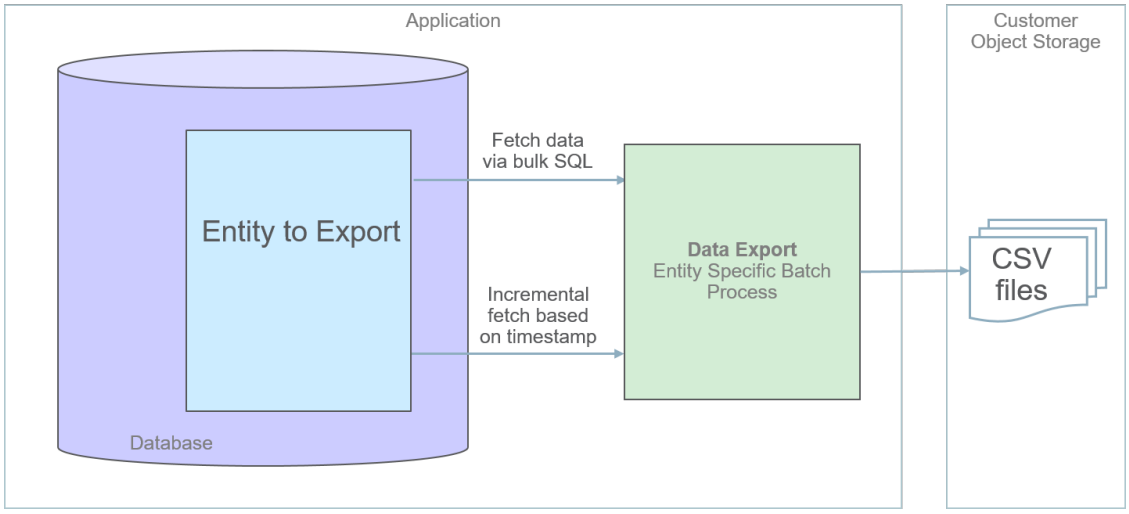
The ongoing export batch process keeps a backup of all entity keys, that were exported on a specific batch run, in a designated backup table. In rare situations where an ongoing export file is deleted or becomes corrupted before it gets processed, the backup table may be used to identify the missing changes and export them again.

Use the Restore Generalized Export Keys (`F1-GERST`) batch process to recover from this situation. The process identifies the entities that were exported during a specified batch run and adds them back to the ongoing changes queue from the backup table. The entities would then be exported again wen the next ongoing export runs. This method does not restore the exact content of the original file that was lost as it cannot be reproduced. Instead, the process ensures that the entities that were included in the lost file would be exported again.

> **Note**
>
> Backup records follow a weekly retention policy for performance reasons. They are meant to support near future situations of file recovery and are not an historical change data capture log.

## Specialized Data Export

Several performance considerations are called for when exporting the data of a high volume entity or an entity that is updated in extremely high frequency. The generalized method does not fit such entities and a specific batch process is designed for each, aiming to achieve an optimal export method for the entity. The initial run of the process exports the entity's existing data while subsequent runs of the same process exports data incrementally, i.e. it exports records that were added or updated since the previous run based a timestamp field. The overall process is illustrated in the following diagram:




The following sections further discusses concepts and guidelines related to the specialized data export method.


Copyright © 2026, Oracle and/or its affiliates.

### A Specific Process For Each Entity

A maintenance object that is designed to be exported using the specialized method is explicitly marked as such by the **Data Export Class** option and its specific batch process is defined using the **Export Batch Control** option.

All specialized batch processes are designed to export the data in a CSV format as it produces smaller files in size compared to more verbose formats. Refer to each batch process for more information about specific export considerations and parameters.

### Enabling Data Export

To enable data export for a maintenance object a specialized type of Data Export Control record needs to be created. The record serves as an indication that such entity should be exported and keeps track of the date and time through which data has been exported so far, allowing subsequent export run to only export incremental changes.

### Incremental Data Export

The specialized export method tracks changes incrementally from prior export run and does not use a change data capture mechanism. It therefore relies on a timestamp field on the primary table of the maintenance object to indicate when data was added or changed. Only entities that were added or changed since the previous export run are exported with each subsequent run. The date and time through which data has been exported is recorded for each maintenance object on its corresponding Data Export Control record.

### Order of Consuming Files

Specialized data export files should be consumed in run number order to ensure a consistent and chronological uptake of data.

# Defining Data Export Control

This portal is used to maintain Data Export Control records. These records capture your organization's export settings and current export state for a selected maintenance object. For a generalized export it keeps an indication as to whether the initial export for the maintenance object has been completed or not and whether changes made to the maintenance object should be tracked and exported on an ongoing basis. For a specialized data export it keeps the date/time through which data was exported.

Refer to Understanding Data Export for more information.

You can access the portal from the**Admin Menu**, and then **Implementation Tools**, and then **Data Export Control.** You are brought to a query portal with options for searching for a specific record. Once a data export control record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Data Export Control** zone provides information about the selected record.

# Generalized Export Dashboard

The number of maintenance objects that are eligible for generalized export is large. This portal allows you to easily select and enable export for many eligible entities at once as well as review the current export state of those already enabled at a glance.


Copyright © 2026, Oracle and/or its affiliates.

Refer to Generalized Data Export for more information about the generalized data export method.

You can access the portal from the **Admin Menu**, and then **Implementation Tools**, and then **Generalized Export Dashboard.**

The following zones may appear as part of the portal's **Main** tab page.

- **Export Entities.** This zone lists entities for which generalized data export control settings have already been set.
- **Eligible Entities.** This zone lists entities that are eligible for the generalized data export method and for which such settings have not yet been set.


Copyright © 2026, Oracle and/or its affiliates.

# Part IV

# Meter Solution Administrative User Guide

Welcome to the Oracle Utilities Meter Solution Administrative User Guide. This document is used with the following products:

- Oracle Utilities Meter Data Management
- Oracle Utilities Smart Grid Gateway
- Oracle Utilities Market Settlements Management
- Oracle Utilities Meter Solution Cloud Service
- Oracle Utilities Market Settlements Management Cloud Service

This section describes administration and configuration features and functionality used with Oracle Utilities Meter solution products.


- Meter Solution Products Functional Overview
- Additional Resources
- Best Practices
- System-Wide Options
- Defining General Options
- Defining Asset Options
- Defining Location Options
- Defining Device Options
- Defining Device Installation Options
- Defining Measurement Options
- Defining Validation, Editing, and Estimation (VEE) Options
- Defining Usage Options
- Defining Time of Use Mapping Options
- Defining Communications Options
- Smart Grid Gateway Adapters
- Defining Service Order Management Options
- Defining Settlement Options
- Additional Independent Modules
- Integrations




Copyright © 2026, Oracle and/or its affiliates.

# Meter Solution Products Functional Overview

### Introduction

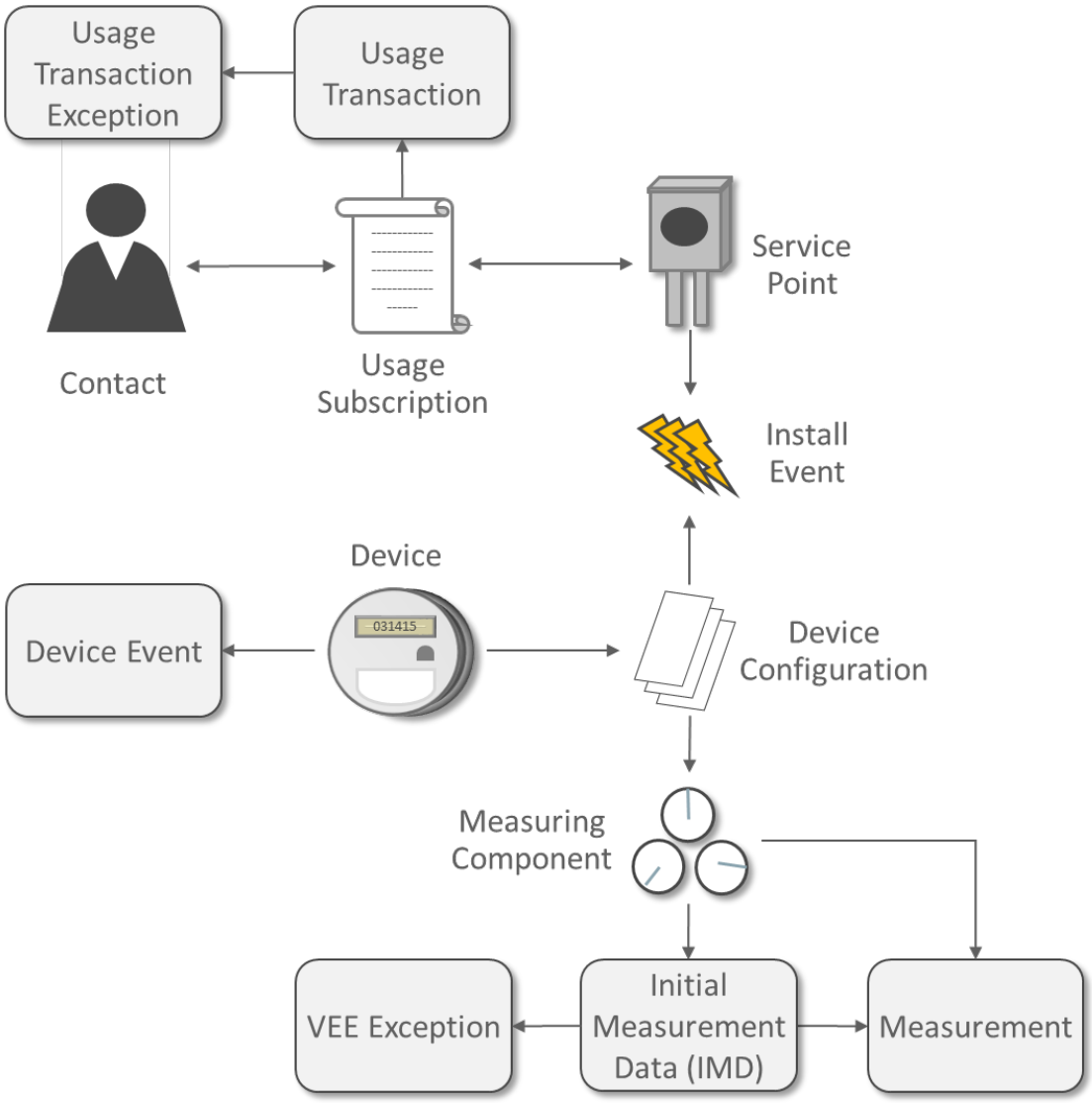
Oracle Utilities meter solution products provide functionality for handling large volumes of meter and device data that enable increased accuracy, flexibility, and scalability. The following outlines the most crucial business processes that are enabled within the system:

- Defining meters, meter configurations, service points, and meter installations.
- Loading meter readings and interval data from a head-end system or other source.
- Automatic validation, editing, and estimating measurement data.
- Robust editing capabilities for readings and interval data
- Calculating and publishing bill determinants, and other data, from measurement data to be used in external, down-stream systems; such as, billing or pricing.

Oracle Utilities meter solution products store a lot of important data; including, meters, service points, customer contacts, and everything in between.


Copyright © 2026, Oracle and/or its affiliates.

**Measurement, VEE, and Usage Calculation for Billing**




A Device represents a physical meter, communication module, or some other device in the field. A Service Point is the point where service is delivered to a customer and a device can be installed. The Install Event is a record of a specific device that is installed at a Service Point. A **Contact** is the customer that is associated with the Usage Subscription.

In order to properly track the way that a device is configured, the Device Configuration keeps a record of which types of data should be measured for the device. A Measuring Component represents a single channel of data for a device; for example, a device may have multiple measuring components: one that represents kWh interval data, another that represents kWh scalar readings, and a third that registers Voltage interval data.

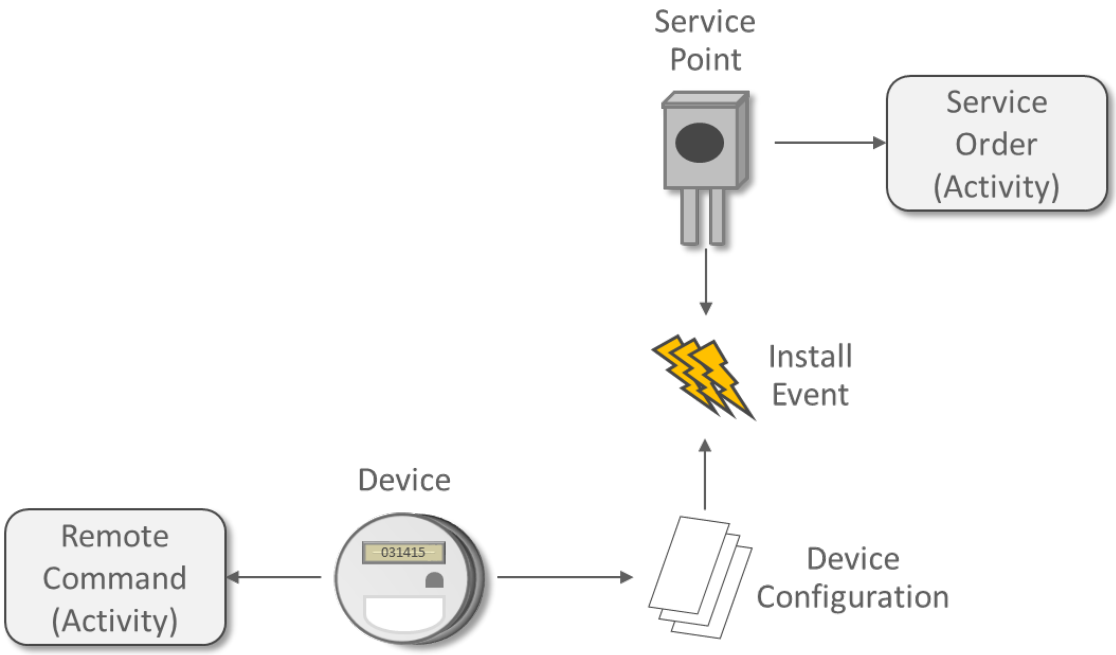
Energy data received from meters is initially stored as Initial Measurement Data (IMD). Once the VEE process is executed, if the data passes then Measurements are created. If the data fails, however, then VEE Exceptions are created.

A Usage Subscription tracks a set of usage calculations that should be performed for a single or multiple Points. In order to perform calculation of usage (often referred to as bill determinants), a Usage Transaction is created through a request from the billing system. If any


Copyright © 2026, Oracle and/or its affiliates.

issues are encountered in the usage calculation process, or from a usage transaction validation, then Usage Transaction Exceptions are created.

**Service Orders and Remote Commands**




An Activity in Oracle Utilities Meter Data Management is a very flexible object that is used for various processes. Two of the key processes handled by Activities are:

- Remote Commands involve communicating with a Head End system to perform actions or retrieve data from a meter; for example, remote connects, remote disconnects, and on-demand reading.
- Service Orders are methods where work can be performed at a Service Point; for example, enabling service, disabling service, and meter exchanges.

Oracle Utilities meter solutions products include the following functional areas:

- **Asset Management** is used by analysts and administrators to manage and define the assets used to record and capture meter data. Optional.
- **Device Management** is used by analysts and administrators to manage and define the devices used to record and capture meter data.
- **Device Installation** is used by analysts and administrators to manage device installation; including, defining markets and service providers, service points, contacts, and installation events.
- **Measurement Data** is a normalized way of storing data from a meter that involves some form of measurement (such as, kWh and CCF). Both interval data and scalar readings are held in this common storage location.
- **Validation, Editing, and Estimation (VEE)** is used by administrators to define validation, editing, and estimation rules to be applied to measurement data. VEE Exceptions may result from validation or estimation failures and should be worked by analysts through the To Do process.


Copyright © 2026, Oracle and/or its affiliates.

- **360 Degree Search and View** is used by analysts and administrators to search and view data for devices, measuring components, service points, usage subscriptions, and contacts.
- **Consumption Sync** provides an automatic method to keep interval data and scalar readings in sync with one another.
- **Usage Calculation** is used by administrators to manage the calculation of usage data and to provide the results of those calculations (commonly referred to as bill determinants) to external systems and parties. Usage calculation groups and rules define calculation rules to be applied to measurement data. Usage Transaction Exceptions may result from usage calculation and should be worked by analysts through the To Do process.
- **Device Events** provide a view of specific events that have occurred on a meter. These are often unexpected and can indicate an issue with the meter.
- **Communication** helps track the instances when communication occurs with external systems. This is heavily used to track remote commands against Head End systems.
- **Aggregations** are used by analysts and administrators to search, view, and maintain aggregated measurements that represent a summarization of other measurements from a set of devices and/or measuring components.
- **Master Data Sync** defines the methods that data is automatically synchronized to Oracle Utilities Meter Data Management from external sources; such as, a CIS and/or Asset Management system.
- **Outage Storm Mode** is a way for Oracle Utilities Meter Data Management to detect widespread outages and suppress estimation for those meters until normal data communication resumes.
- **Dashboards** provide high level metrics for Oracle Utilities Meter Data Management operators to monitor operational trends as well as overall system health.
- **Service Issue Monitor** can be setup by administrators to monitor various conditions within Oracle Utilities Meter Data Management and automatically create a Service Investigative Order, if those conditions are met.
- **Service Order Management** provides a centralized program for managing the complex interactions required for Service Order processing. This area is especially valuable for managing service order processing that involves remote communication with a Head End for connects, disconnects, and on-demand readings.
- **Market Settlement** provides the core functionality for calculating and settling energy changes.
- **Information Lifecycle Management** is an automated method that administrators can configure to prepare data for archiving or purging after a defined period of time for the record type.


- Architectural Overview
- Naming Conventions
- High Level Functional Areas
- Meter Data Management Configuration Setup Sequence
- Operational Device Management Table Setup Sequence
- Understanding Online / Batch Loaders


# Architectural Overview



Copyright © 2026, Oracle and/or its affiliates.

Oracle Utilities meter solution products are used to maintain information about meters and the service points at which they are installed. The applications provide means of recording measurements and events associated with meters in the field as well as the ability to compute usage for the recorded measurements.

Oracle Utilities meter solution products comprise the following functional areas:

- Asset Management: the maintenance of physical assets in the field
- Device Management: the maintenance of physical meters in the field
- Device Installation: the maintenance of service points and the installation of meters in the field. This includes the means of registering outside systems to Oracle Utilities Meter Data Management for provider/consumer-specific processing of meter events and activities
- Device Communication: the maintenance of communications between Oracle Utilities Meter Data Management and head-end systems, including import of usage and events.
- Validation, Editing, and Estimation: the maintenance of measurement data and the engine used to validate and modify that data as it comes in
- Usage Management: the engine that calculates billable usage recorded on devices, applying factors and dividing the usage into configurable time of use periods
- Aggregation: summarization of measurement data for reporting and analysis purposes
- Data Synchronization: synchronizing data between Oracle Utilities Meter Data Management and other Oracle Utilities applications (including Oracle Utilities Operational Device Management and Oracle Utilities Customer Care and Billing).
- Oracle Utilities DataConnect: extraction of consumption and master data for use in external systems.
- Settlement: calculation and reconciliation of energy charges within a market

Oracle Utilities Meter solution products are built upon the Oracle Utilities Service and Measurement Data Foundation, a framework that provides shared functionality used by Oracle Utilities Meter Data Management, the Oracle Utilities Smart Grid Gateway adapters, and other Oracle Utilities products. Oracle Utilities Meter Data Management and the Oracle Utilities Service and Measurement Data Foundation are built atop the Oracle Utilities Application Framework.

# Naming Conventions

Oracle Utilities Meter solution products use naming conventions to identify and distinguish entities that belong to different Oracle applications. These conventions can help you locate entities and understand their context.

Each base product uses a 2-character prefix for all its entities. For Oracle Utilities Meter solution products, these prefixes are as follows:

| Product Name | Prefix |
|---|---|
| Oracle Utilities Application Framework | F1 |
| Oracle Utilities Service and Measurement Data Foundation | D1 |
| Oracle Utilities Meter Data Management | D2 |
| Oracle Utilities Smart Grid Gateway for Landys+Gyr | D3 |
| Oracle Utilities Smart Grid Gateway for Echelon | D4 |


Copyright © 2026, Oracle and/or its affiliates.

| | |
|---|---|
| Oracle Utilities Smart Grid Gateway for MV-90 | D5 |
| Oracle Utilities Smart Grid Gateway for Sensus RNI | D6 |
| Oracle Utilities Smart Grid Gateway for Silver Springs Networks | D7 |
| Oracle Utilities Smart Grid Gateway for Itron OpenWay | D8 |
| Oracle Utilities Smart Grid Gateway for Adapter Development Kit | DG |
| Oracle Utilities Customer Self Service | WX |
| Oracle Utilities Meter Data Management Demo Data* | DM |

*Note: this is not actually a product but rather a set of data that demonstrates Oracle Meter Data Management functionality.

Oracle recommends that you follow these naming conventions and develop your own set of conventions for the entities you create. If you create new entities, DO NOT use these prefixes; use the prefix "CM" (or some other unique prefix) to identify entities that have been customized.

# High Level Functional Areas

A high level overview of the functional areas for Oracle Utilities Meter solution products has been defined in the Functional Overview section of the *Oracle Utilities Meter Solution Business User Guide*. This may be worthwhile to review to become better oriented to the areas of the Administrative User Guide before reading on.

# Meter Data Management Configuration Setup Sequence

This section provides the suggested order for the setup of Meter Data Management and Smart Grid Gateway administrative data.

| | Entity | Description |
|---|---|---|
| 1 | Feature Configuration | These are options that can be set to impact system processing. Typically these have wide ranging impact across several modules (e.g., multi-time zone support) |
| 1 | Country | Your organization's country. |
| 1 | Currency | Your organization's native currency. |
| 1 | Display Profile | Controls how dates, times, and numbers displayed. |
| 1 | Language | The language to use for this implementation. |
| 1 | Time Zone | Your organization's base time zone. |


Copyright © 2026, Oracle and/or its affiliates.

| | Entity | Description |
|---|---|---|
| 1 | Work Calendar | The work calendar for your organization, which identifies your public holidays. |
| 2 | Installation Options | Control various global aspects of the system. |
| 2 | To Do Role | Used to associate users with To Do entries. |
| 2 | To Do Type | Used to define types of To Do Entries. |
| 3 | User | Defines a user's user groups, data access roles, portal preferences, default values, and To Do roles. |
| 3 | User Group | A group of users who have the same degree of security access. |
| 3 | Service Type | Defines specific types of service for which usage can be recorded and captured (electric, gas, steam, etc.). |
| 4 | Division | Delineates between different operating companies within a large conglomerate of utilities. |
| 4 | Unit of Measure | Quantities measured and recorded by the system (CCF, KWH, KW, etc.). |
| 4 | Service Quantity Identifier | Used to further distinguish between measured quantities that have identical UOM/TOU combinations. |
| 4 | Time of Use | Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used (On-Peak, Off-Peak, etc.). |
| 4 | Factor | Centrally stored sets of values for use in validation rules, bill determinants calculations, and other processes. |
| 4 | Market | Defines jurisdictions or regulatory environments in which a Service Point participates. |
| 4 | Measurement Cycle | Defines the schedule for manual meter reading of devices at Service Points in that cycle |
| 4 | Measurement Cycle Schedule | Define the dates on which devices are scheduled to be read for a given measurement cycle. |
| 4 | Outbound Message | Defines messages sent to external systems. |


Copyright © 2026, Oracle and/or its affiliates.

| | Entity | Description |
|---|---|---|
| 4 | Processing Timetable Type | Defines types of schedules that can be referenced by different processes and objects. |
| 4 | IMD Type | Define how initial measurements are processed using Direct Measurement Processing. |

**Note**

The base package includes several pre-defined IMD types that can be used when processing data from different head end systems supported through Smart Grid Gateway Adapters.


Copyright © 2026, Oracle and/or its affiliates.

| | Entity | Description |
|---|---|---|
| 5 | External System | Defines External Systems with which Oracle Utilities Meter Data Management should be able to communicate. |
| 6 | Head End Systems<br>External Applications<br>Market Participants | External entities that serve various roles relative to the application (head-end systems, billing systems, market participants, outage management systems, etc.). |
| 6 | Contact Type | Defines properties of a class of entities (businesses, persons). |
| 7 | Activity Type | Defines properties common to a specific type of activity. |
| 7 | Communication Type | Define properties common to a specific type of communication. |
| 7 | Service Task Type | Defines specific types of tasks performed by external users (self-service meter reads, self-service outage notifications, etc.) |
| 7 | Dynamic Option Type | Defines information common to dynamic options of a specific type. |
| 7 | Manufacturer | Individual companies that makes devices. Manufacturers also reference models. |
| 7 | Exception Type | Defines properties common to VEE Exceptions of a specific type. |
| 8 | VEE Group | Collections of VEE Rules that are applied to initial measurement data. |
| 9 | VEE Rule | Standard and custom VEE Rules that perform checking and/or manipulation of initial measurement data. |
| 10 | VEE Eligibility Criteria | Dictates whether a VEE Rule can execute based on a set of defined criteria. |
| 10 | Device Event Type | Defines information about a class of device events. |
| 11 | Measuring Component Type | Defines the most important properties of a measuring component. |
| 11 | Measuring Component Comparison Type | Defines details by which measuring component data can be compared to determine the days that most closely resemble a specific day being evaluated. |


Copyright © 2026, Oracle and/or its affiliates.

| | Entity | Description |
|---|---|---|
| 12 | Device Configuration Type | Defines properties of Device Configurations of a given type. |
| 13 | Device Type | Defines information about a class of devices. |
| 14 | Service Point Type | Defines specific types of points at which service is delivered. |
| 14 | Service Point Quantity Type | Defines types of quantities that can be stored for a service point. |
| 14 | Usage Subscription Quantity Type | Defines types of quantities that can be stored for a usage subscription. |
| 14 | Entity Group | Defines groups of entities, such as devices (including meters or items, as well as controllable devices used with Customer Program Management Cloud Service and Digital Asset Management) or service points. |
| 15 | TOU Group | Used to limit the set of Time Of Uses that are usable in a TOU schedule. |
| 16 | TOU Map Template | Schedules used for TOU map data generation. |
| 17 | TOU Map Type | Define important properties of TOU maps of a given type. |
| 18 | Usage Transaction Exception Type | Defines properties common to Usage Transaction Exceptions of a specific type. |
| 18 | Usage Calculation Group | Collections of usage calculation rules that are applied to measurement data to calculate bill determinants for Usage Subscriptions. |
| 19 | Usage Calculation Rule | Defines rules that perform calculations on measurement data to generate bill determinants and other values used by external systems. |
| 20 | Usage Calculation Rule Eligibility Criteria | Dictates whether a usage calculation rule can execute based on a set of defined criteria. |
| 21 | Usage Subscription Type | Defines collections of properties common to a set of Usage Subscriptions. |


Copyright © 2026, Oracle and/or its affiliates.

| | Entity | Description |
|---|---|---|
| 22 | Data Source | Defines the source of data for dynamic aggregation, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or measuring component sets |
| 22 | Dynamic Aggregation Measuring Component Type | Define the most important properties of a measuring component used with dynamic aggregation. |
| 22 | Measuring Component Set | Define the dimensions and criteria by which dynamic aggregation will be performed. |
| 22 | Aggregation Group | Define the ordering of a series of related aggregations based on a set of configured measuring component Sets. |
| 23 | Settlement Subscription Type | Defines collections of properties defining a class of settlement subscriptions. |
| 23 | Settlement Unit | Defines the “lowest common denominator set” of dynamic aggregation dimensions. |
| 23 | Settlement Calculation Group | Defines collections of settlement calculation rules that are used to perform settlement calculations, including calculating service point quantities, totaling consumption and usage for settlement units, application of losses, and allocation of unaccounted for energy (UFE). |
| 23 | Settlement Calculation Rule | Defines rules that perform settlement calculations, including calculating service point quantities, totaling consumption and usage for settlement units, application of losses, and allocation of unaccounted for energy (UFE). |
| 23 | Settlement Item Type | Defines specific types of end-customer settlement accounts. |
| 23 | Market Contract Type | Defines specific types of market contracts used in settlement processing. |
| 23 | Market Product Set | Defines the highest level grouping for a set of market products that will be processed together. |
| 23 | Market Product Type | Define types of products used in settlement processing. |


Copyright © 2026, Oracle and/or its affiliates.

| | Entity | Description |
|---|---|---|
| 24 | Processing Methods | Control various behaviors for external applications, head end systems, and market participants within the system such as which message is sent, how an external value is translated, among others. |
| 24 | Master Configuration | Configuration that applies to series of modules that acts as a central point of configuration rather than embedding repetitive configuration throughout a set of algorithms. |
| 24 | Information Lifecycle Management | Information Lifecycle Management (ILM) is designed to address data management issues, with a combination of processes and policies so that the appropriate solution can be applied to each phase of the data's lifecycle. |
| 24 | Analytics | Defines the extract parameters, the bucket configurations and configuration snapshots used for extracting data for Oracle Utilities Analytics |

# Operational Device Management Table Setup Sequence

This section provides the suggested order for the setup of Operational Device Management administrative data used with Asset Management features and functions.

> **Note**
>
> Asset Management features and functions are disabled by default. To enable these features, execute the Turn On ODM and C2M (D1-OnOdmC2m) Business Process Assistant (BPA) script.

| Sequence | Entity | Description |
|---|---|---|
| 1 | Configuration Type | Defines the information required for various asset configurations with other assets, components, firmware, and other data. |
| 1 | Service History Questions | Used to guide workers through corrective and preventive inspection processes as part of an asset's service history. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| 1 | Service History Type | Defines the main attributes for service history of a certain type. Types might include maintenance, inspection, service, test, measurement, and so on. |
| 2 | Asset Location Type | Defines the details of asset or component locations of a certain type. Valid types of asset locations can include service points, other locations for installed assets, or any other location as defined by your organization. |
| 2 | Out of Service Location Type | Defines attributes of types of locations where assets that are out of service are stored and maintained. Examples of out of service locations include warehouses, storerooms, and repair shops. |
| 2 | Organization | Used for categorization and reporting for locations. |
| 2 | Configuration | Documents and manages valid setups for assets and components. This includes a specification for each asset or component in the configuration as well as the valid value for each parameter, such as the firmware version of each asset. |
| 3 | Out of Service Location | Defines physical locations where assets are stored or repaired such as a storeroom, warehouse, or service facility. When assets are changed to statuses such as “In Repair” or “Pending Disposition” the asset location is typically moved to an out of service location. |
| 3 | Configuration Reports | Provides a summary of currently installed configurations. |
| 4 | Asset Type | Defines the attributes for assets and components of a certain type, including information such as the valid location types and service history types for the assets and whether or not such assets can have attached components.<br><br>When setting up Asset Types, asset type codes should match corresponding device type codes in Meter Data Management (see Meter Data Management Table Setup Sequence). |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| 5 | Asset Activity Type | Defines the attributes for asset activities of a certain type, including scheduling information , exception handling, attachments, and related completion event types.<br><br>**Note**<br>Asset activity types are automatically created when Service Order Management Field Task Types are created (values for the Field Task Type extendable lookup). |
| 5 | Asset Completion Event Type | Define various processes or events to be executed when completing an asset activity such as asset replication, service history creation, attachment creation, and activity creation. |
| 5 | Asset Inbound Communication Type | Defines the information required for the system to receive messages from external systems. |

# Understanding Online / Batch Loaders

Online / Batch Loaders enable the quick loading of data into the system. Data can be loaded for any business object as it uses the business object's schema to load data into the application.

Note: We recommend that this process not be used to load data for objects maintained through merged maintenance, such as Service Type, Unit of Measure, Service Quantity Identifier, Time


Copyright © 2026, Oracle and/or its affiliates.

of Use, Service Point Type, or Service Point as this process may not properly populate data for all maintenance objects supported by merged maintenance.

Two types of loaders are supported:

- **Online Loaders** easily upload smaller amounts of data. Use this type of upload process when uploading small numbers of records or data for business objects without complex lifecycles and business object rules such as Organizations. The data is loaded through the user interface to create the target objects.
- **Batch Loaders** should be used for larger volumes of data and for complex business objects such as Assets. Batch loaders load data from files on the server or attachments. Batch loaders use a two-step process that first creates the inbound sync request and uploads the file to a staging table and secondly, uploads the sync request from the staging table to either create the target objects or create another dedicated inbound sync request that handles the complexity of a particular object.

Processing uses the creation of a request type to facilitate the upload process:

- For online request types, the actual data is stored within and processed by the request.
- For batch request types, two options for batch load are supported; from a file located on the server or from an attachment in the application. Either the file name (file on server) or the attachment id (attachments) is stored on the request, the data is loaded into the generic inbound sync requests, and the request acts as a header record for all the generic inbound sync requests.

A csv file is the format used for loading data (see **Creating CSV Files**, below). The loader has very strict requirements as to the format of the file. Please see the embedded help in the **Data** section on the Online Loader Request business object for details and additional tips.

- Creating CSV Files
- Initiating Requests

# Creating CSV Files

While data can be manually entered into the system, the most efficient method for uploading the data is to use one of the CSV data templates already available for all appropriate business objects.

The first step is to create a .csv file (comma delimited) from the business object of the portal being loaded.

The following is an example using Organization (W1-Organization):

1. Navigate to **Admin**, select **Location**, select **Organization**, and then select **Search**.
2. Search for and select a record.
3. Expand the **Record Information** section and click the "Organization" **Business Object** link to navigate to the **Business Object** portal.
4. Click the **View Schema** link on the **Main** tab and then copy the elements from the business object to a CSV file to create your data.

| 1 | Sequence | Primary Key | Field 01 | Field 02 | Field 03 |
|---|---|---|---|---|---|
| 2 | 1 | Organization | Organization Type | Disposition | description |
| 3 | 2 | nodeId | nodeType | nodeDisposition | description |


Copyright © 2026, Oracle and/or its affiliates.

| 1 | Sequence | Primary Key | Field 01 | Field 02 | Field 03 |
|---|---|---|---|---|---|
| 4 | 3 | | Group | W1AC | Group 1 |
| 5 | 4 | | Group | W1AC | Group 2 |

Notes:

- Columns in the top row match the elements in the list of the request business object (header).
- The row with the lowest sequence number (1) is the description of xpaths to help users read the file and is ignored by the processing.
- The business object processing treats the row with the second lowest sequence number (2) not as data, but as the xpath to move data to the target business object, in this case, W1-Organization.
  - Note that the configuration of some business object mean that this cell may start with a “+” sign. Also note that the file always treats this cell as a formula and adds an “=” sign to the front of the data string, as indicated by “#NAME?” (For example: =+nodeId/nodeType). You must reformat the cell to “Text” and remove the “=” sign.
- Rows with sequence number three (3) and up are treated as data.
- For objects where the primary key is a system generated ID (such as the case with Organization), the Primary Key column cells blank.

Note: When adding records that reference look up values use the lookup field value, not the description. For instance, the Disposition field references the NODE_DPOS_FLG lookup, for which the "Active" field value is "W1AC". For fields that reference other objects, use the code, not the description.

Other Formatting Issues

The following information addresses known formatting issues in working with the CSV data template:

- Dates need to be formatted as yyyy-mm-dd (2020-10-27).
- Date and Time should be formatted as yyyy-mm-dd-hh.mm.ss.
- Columns based on lookup values that have leading zeroes will need to match the values in the CSV file.
- A link needs to be created when you want to include the “parent”, which has a system generated ID, in the same file with the “child”. For example, if you want to populate the Parent Organization’s ID in the same file, you would need to insert a link (for example, @3, which is the @ symbol plus the Sequence #) in the row where the system generated ID would reside.
- If multiple rows contain the same Primary Key, only the last row is loaded. All values for the Primary Key need to be on the same row.

# Initiating Requests

There are two steps involved in loading data using this process: creating the request type and the request.


Copyright © 2026, Oracle and/or its affiliates.

### Creating Request Types

1. Navigate to **Admin**, select **General**, select **Request Type**, and then select **Add**. The **Select Business Object** screen opens.
2. Select the request type business object that corresponds to your request type (“Online Loader Request Type” or “Batch Loader Request Type”) from the **Request Type Business Object** drop-down list and click **OK**. The **Online Request Type** or **Batch Request Type** screen opens (based on the business object selected).
3. Enter a code and meaningful description for the request type in the **Request Type** and **Description** fields. For example, "Online Loader Request Type" or "Batch Loader Request Type".
4. Enter a **Detailed Description** (optional).
5. For request types based on the Batch Loader Request Type business object, specify whether or not the contents from the attachment record will be deleted once the upload is finished (Yes or No) in the **Deleted Attachment Upon Finish** drop-down list.
6. Click **Save**.

### Creating Requests

#### Online Requests

1. Navigate to **Menu**, select **Tools**, select **Request**, and then select **Add**. The **Select Type** screen opens.
2. Select your online loader request type and then click **OK**. The **Online Loader Request** screen opens.
3. Select the business object for your loader in the **Business Object** field.
4. Select your CSV file to upload or enter the data directly into the Data section and then click **Save**. Note: If examples of online requests are already available in your system, search for the appropriate request, click **Edit**, and either use the existing data if it applies, or click **CSV File To Upload** to upload your already formatted and populated CSV file. You can also click **Download To CSV File** and manually populate the data into the file and then click **CSV File To Upload** to upload the file. Your request will open in the **Request** zone in the "Created" state.
5. Click **Load Data** in the **Record Actions** section to submit the request. If successful, the status of the request will be "Loaded". Check the **Request Log** for requests that do not transition to a “Loaded” status. You will need to click **Back to Created** to reset the request back to the "Created" status and click **Edit** to fix the issue before resubmitting the request.

#### Batch Requests

1. Navigate to **Menu**, select **Tools**, select **Request**, and then select **Add**. The **Select Type** screen opens.
2. Select your batch loader request type and then click **OK**. The **Batch Loader Request** screen opens.
3. Select an external system from the **External System** drop-down list.
4. Select the business object for your loader in the **Business Object** field.
5. Enter a commit frequency in the **Commit Frequency** field.
6. Enter the **File Name** or search for and select the **Attachment** that should be in the request and click **Save**.


Copyright © 2026, Oracle and/or its affiliates.

- For **Batch File Upload** — The Filename is specified in the request. Comma delimited files should uploaded to or available on the server. Run the W1-GBUPL batch job to create the sync request. Parameter File Name and File Path should match the file name and path of the file. Once the job has ended and the request is loaded, run the F1-SYSRQ batch job to process the created sync request.
- For **Batch Attachment Upload** — The Attachment ID is specified in the request. Comma delimited file is uploaded as an attachment. Run the W1-GAUPL batch job to create the sync request. Parameter Request ID should match the created request. Once the job has ended and the request is loaded, run the F1-SYSRQ batch job to process the created sync request.
- Use the following parameters when running the F1–SYSRQ batch job:
  - Maintenance Object: F1–SYNCREQIN
  - Restrict by Batch Code: True
  - Restrict by Business Object: W1–GenericDataLoaderSync
  - Restrict by Status Code: PENDING

# Additional Resources


- How to Get Support
- Knowledge Base Articles
- Support Hot Topic Emails
- Embedded Help
- Leveraging Demonstration Data
- Customer User Groups
- Oracle Forums


## How to Get Support

Oracle is deeply committed to technical support services for its products. The Oracle Support online portal provides a reliable, easy-to-use method for obtaining technical support for Oracle Utilities Meter Data Management. To register as a new user for Oracle Support, go to https://support.oracle.com and follow the steps shown on the screen.

Oracle policies for standard technical support services are available at http://www.oracle.com/us/support/policies/index.html.

## Knowledge Base Articles

Once you have access to Oracle Support, finding Knowledge Base articles is an easy process.

1. Log in to https://support.oracle.com.
2. Select the **Knowledge** tab at the top of the Oracle Support portal.
3. Under the Knowledge Base section, type "**Oracle Utilities Meter Data Management**" as the product.



Copyright © 2026, Oracle and/or its affiliates.

4. Enter key search topics into the **Enter search terms** box.
5. Click **Search**.

The search results will display all knowledge articles stored on Oracle Support related to Oracle Utilities Meter Data Management for your key topics.

- Important Articles

## Important Articles

While there is a vast library of important Knowledge Base articles on Oracle Support, the list below highlights a few of the key knowledge articles that should be helpful if you're looking for additional detailed information about Oracle Utilities Meter Data Management:

| Article Name | Knowledge Base Article Number |
|---|---|
| Related Measuring Component Consumption Sync Technical Reference Paper | KB406197 |

# Support Hot Topic Emails

Oracle Support provides a way to receive daily or weekly updates on new information posted for Oracle Utilities Meter Data Management.

1. Log in to https://support.oracle.com.
2. Select the **Settings** tab at the top of the Oracle Support portal. (Note: this tab may be hidden for you and may require clicking a dropdown on the right side of the tabs)
3. Once on the Settings screen, choose the **Hot Topics Email** from the set of options on the left.
4. Under Delivery Options, choose **Send With Selected Options**.
5. Choose your delivery frequency: either Daily or Weekly.
6. Add content which interests you: there are a number of options available. To subscribe to updates for Oracle Utilities Meter Data Management, hit the **Add...** button under Selected Products. Enter "Oracle Utilities Meter Data Management" for the Product then choose the types of information for which you'd like to receive an email.
7. Make sure you click **Apply** at the bottom once you've made all of your selections.

# Embedded Help

Oracle Utilities Meter Data Management, like all Oracle Utilities Application Framework applications, provides extensive internal documentation. For example, detailed descriptions of system objects are included in the objects' maintenance portals. The lifecycle of each business object is described on the Lifecycle tab and depicted in flow diagrams on the Summary tab. This information is extremely useful for implementers and system administrators.

Embedded help is provided for all non-obvious fields in most portals and zones. If a field has associated help text, a question mark icon appears next to the field when the zone is displayed.


Copyright © 2026, Oracle and/or its affiliates.

# Leveraging Demonstration Data

One of the best ways to understand Oracle Utilities Meter Data Management is through a thorough review of the demonstration data (also referred to as "demo data") provided by Oracle. The demo data for Oracle Utilities Meter Data Management provides the following:

- An overall view of how to implement the product for common needs
- Examples of the productized solutions to solve key utility scenarios
- Some examples of adding "customer modifications" on top of Oracle Utilities Meter Data Management

This demo data can be downloaded along with the Oracle Utilities Meter Data Management. It's recommended that this be installed into an isolated environment for reference purposes separate from other development or testing environments.

Once installed, the demo data scenarios can be found in two places within Oracle Utilities Meter Data Management:

- A zone named **Demo Scenarios** is shown in the Sidebar. This provides a quick method to link to the 360 Degree View for the specific demo meter.
- An extendable lookup named **Demo Scenarios Catalog** describes each of the scenarios in detail. This matches the scenarios displayed in the Demo Scenarios zone.

**NOTE:** Oracle recommends that you do not clone the demonstration environment as a basis for a new production environment. The demonstration environment typically includes transactional data that will be irrelevant to your production environment and can cause unexpected issues if it is not purged correctly. The recommended process is to start a new production environment from a new installation and migrate "clean" system data (such as business objects and algorithms) and administrative data (such as sample activity types or other administrative entities) from the demonstration and/or test or development environments as applicable. Instructions for using these tools are contained in the Bundling and Content Migration Assistant sections in the Oracle Utilities Application Framework *Administrative User Guide*.

# Customer User Groups

A number of Customer User Groups have been established (one in the United States and others internationally) that are specific to sharing best practices and learnings about Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway (SGG). If you're interested in participating in a Customer User Group, contact the appropriate Product Manager.

# Oracle Forums

Oracle Forums is a way for you to suggest ideas for Oracle Utilities solutions. Oracle Forums provides an easy way to submit feature ideas directly to product owners in Oracle Utilities. Once an idea is submitted, other users can comment on and vote for your idea to show their support.

Submitting an idea is simple. When you go to Oracle Forums, click **Sign In** to enter your Oracle account information or create a new account. Once you have signed in, select **New Post.** Select a solution category, title, and description for your idea. Click **Add Post** and product owners will be notified of your suggestion.


Copyright © 2026, Oracle and/or its affiliates.

# Best Practices


- Performance Recommendations
- Data Access and Retrieval Recommendations
- Referencing Master Data by Identifiers
- Smart Grid Gateway Native Implementations
- Recommendations for Creating an On-Premises Production Environment


## Performance Recommendations

This section includes best practices and recommendations regarding performance.


- Partitioning and Data Removal
- Device Channels to Process
- Data Conversion
- Initial Measurement Loading Recommendations
- VEE Recommendations
- Measurement Processing Recommendations
- Aggregation Recommendations
- Usage Transaction Recommendations
- Usage Calculation Recommendations
- User Interface Recommendations
- Exception Handling Recommendations
- SQL Recommendations
- Custom Rules and Processes
- Testing Recommendations
- Java Recommendations — On-Premises Implementation Only


### Partitioning and Data Removal

- **Implement Partitioning**: Partitioning is essential to manage the large amounts of data coming into Meter Data Management. Customers will have significant performance problems without partitioning. Partitioning is required for good performance and to organize the data so old data can easily be dropped from the database. Customers need historic partitions (for conversion and replacement reads) and future partitions.
  - Cloud Implementations: Customers are no longer required to run the "Add Partition" (K1-ILMAD) batch process every month. Oracle automatically adds partitions for entities, every month, on desired domains and schemas. However customer are required to run the "ILM Automation - New Subretention Value" (F1-ILMSV) batch process (for sub-partitions, if needed) on a regular basis to ensure that new partitions are created as needed.



Copyright © 2026, Oracle and/or its affiliates.

  - On-Premise Implementations: Please refer to the *Oracle Utilities Meter Data Management Database Administration Guide* for partitioning guidelines for on-premises implementations.

- **Implement Compression**: Compression is essential to reduce the size of the database. For standard hardware we recommend Advanced Compression. For engineered systems that use Oracle Exadata, we highly recommend HCC (Hybrid Columnar Compression) this reduces the overall database by approximately 50%.
  - Cloud Implementations: Customers should use the “ILM Automation – Non-Measurement Compression” (K1-ILMNC) and “ILM Automation – Measurement Compression” (K1-ILMMC) batch processes to change the compression level from Advanced to Hybrid Columnar for non-Measurement data at least two months old, which can result in significant disk space savings. Use the following parameters when running this batch process: Owner, MO (optional), DOP (Degree of Parallelism), Estimated Time Limit (How long job can run until it will not compress another partition). In larger databases this can run for several days a month.
  - On-Premise Implementations: Please refer to the *Oracle Utilities Meter Data Management Database Administration Guide* for compression guidelines for on-premises implementations.
- **Setup Information Lifecycle Management (ILM)**: – The Information Lifecycle Management processes mark items eligible to be removed and update partitions to segregate the data that can be removed. Customers need to define the “ILM Configuration” default retention periods (via the ILM Master Configuration). The retention period should be defined for all ILM objects. ILM Crawlers are provided for low volume transactions. High Volume transactions are marked as eligible for archiving automatically.

  > **Note**
  >
  > Completed IMDs (Initial Measurement Data records) can be removed shortly after being imported. Removing IMDs after 3 months will reduce the compressed database by approximately 50%. There will be database storages savings by archiving other ILM objects as well, such as Activities and To Do Entries.

  Different types of implementations call for different approaches to using ILM.
  - Cloud Implementations: Run the ILM Crawlers Batch Job (batch controls by object) and once the data is aged, run the "ILM Automation - Drop Partition" (K1-ILMDR)_batch process to remove the old partitions from the cloud service database. Please refer to the detailed description of this batch process for additional details.
  - On-Premise Implementations: Please refer to the *Oracle Utilities Meter Data Management Database Administration Guide* and Information Lifecycle Management for ILM guidelines for on-premises implementations.

# Device Channels to Process

Oracle does not recommend customer process all channels of data that Meter Data Management can derive.

- Oracle does not recommend utilities hard-code devices with time of use periods (e.g.: on-peak, off-peak). Hard-coded time of use periods will likely require reprogramming in the future when those periods change. Meter Data Management can easily calculate time of use values from interval data.


Copyright © 2026, Oracle and/or its affiliates.

- Oracle does not recommend customers hard-code demand channels (such as channel used to measure kW), since these must be reset every bill cycle. Demand can be derived from kWh interval data.
- Likewise, Oracle does not recommend customers hard-code channels for Power Factor since that can be derived from kWh and kAVRh interval data.

# Data Conversion

- Only convert essential data. Only convert objects that are essential to running the business.
- Pay careful attention to measurement data. Measurement data is very, very large and may take weeks to convert. Consider this: if it takes one hour to load measurements every day, how long will it take to load 5 years of measurements? Although conversion jobs are faster than regular data loading, measurement conversion still takes time. Here are a few measurement best practices:
  - Projects are encouraged to only convert the minimum amount of measurement data required.
  - If historical measurements are required, try to limit the amount of data converted. The best practice is to convert 13 months of consumption to support VEE.
  - Only convert channels absolutely required, usually billing channels, and don't convert other interval data if not needed.
  - Measurement data values should be converted directly into the Measurement table and not use the IMD and VEE processes.
  - **Do not convert Initial Measurement (IMDs)**. Completed IMDs are not needed for any Meter Data Management processes.
  - Remember that billing quantities will likely also be stored in the customer information system, so confirm that the measurement data is required before conversion.

# Initial Measurement Loading Recommendations

**General Recommendations**

- We recommend using Direct Measurement Processing for initial load IMDs, this has a performance boost over legacy IMD initial load processing while also saving considerable storage.
- We recommend using the SGG native implementation payload upload.
  - SGG native usage upload is most efficient when it is processing files of roughly equal size where the number of files is evenly distributed across the number of threads used for processing. For example, if you have 10 threads processing you would want at least 10 roughly equal sized files.
    - If your head end system does not produce enough files or the files vary greatly in size then we recommend implementing a process external to MDM to split the files so they are (1) roughly equal in size and (2) evenly distribute across the number of threads used for processing
      - If you are unable to implement an external process to split the files then certain adapters support a chunkSize parameter that can be used to ensure the files


Copyright © 2026, Oracle and/or its affiliates.

are evenly distributed across the threads. See Common Parameters for more information about the `chunkSize` parameter.

- Load initial measurements and perform VEE processing every 4 to 6 hours to optimize performance.
- Initial measurement data (IMD) processing is impacted by a number of factors, including the size of the IMDs being processed, and other concurrent processing (such as other batch processes).
  The size of an IMD is determined by two primary factors, the interval size and the duration of the IMD.
  - The interval size determines how many measurement values are contained in the IMD. For example, an IMD containing 1 days' worth of measurement of hourly intervals contains 24 interval values, while an a IMD containing 1 days' worth of measurement of 15-minute intervals contains 96 interval values, and an IMD containing 1 days' worth of measurement of 5-minute intervals contains 288 interval values. All three contain 1 days' worth of measurements but will require different amounts of compute processing.
  - The second factor impacting the size of an IMD is its duration. However, as noted above the interval size of the IMD plays a significant role in determining the overall size of the IMD. As a general rule, the smaller the interval size, the shorter the duration should be.

  We recommend that you limit the number of measurements in a single IMD. In general we recommend a maximum of 1 days' worth of measurements in a single IMD, but this should be adjusted based on the interval size (again, the smaller the interval size, the shorter the duration should be). We recommend trying different combinations of durations and the number of IMDs (such as 3 IMDs per day each containing 8 hours of intervals, 4 IMDs per day each containing 6 hours of intervals, and so so) to find a volume that works best in your environment.

  One other factor to consider if using legacy measurement processing is the impact on database storage. In general, the number of IMDs has a greater impact on database storage than the number of intervals within each IMD – sending more IMDs per day results in a larger impact on database storage. For example, sending the same number of intervals in a day in a single IMD has a smaller database storage impact than sending the same number of intervals spread across multiple IMDs.

  Mitigating concurrent processing involves optimizing the scheduling of IMD and other processing to best suit the needs of your implementation. The initial measurement data and event loading processes should be scheduled to avoid overlap with other intense processes. For instance, do not produce bill determinants when loading meter readings.
- Initial measurement payloads should have very selective criterion to get to the exact measuring component (MC). In an ideal case, it should be the MC identifier along with device serial number. If this is not provided, then the UOM/TOU/SQI configured on the MC type is used to retrieve the exact business object (BO) from the Service Provider which might have the same values for multiple channels.
  Typically, projects use tools such as SQL Loader to migrate large volumes of historical data directly into the Measurement table rather than use IMDs.
- Avoid loading large numbers of initial measurements for a single device configuration in a single payload.
  - D1-IMDV2 helps maintain a constant throughput of IMDs, however, a large number of IMDs for a single device configuration will still cause the thread that picks up that work complete materially later than other threads thus delaying your batch schedule
- Populating the raw data section of the IMD record will reduce overall throughput. Avoid retaining "raw" data in Smart Grid Gateway payload processing whenever possible.


Copyright © 2026, Oracle and/or its affiliates.

- Payload processing (in which usage and event data are loaded into the system) should be run separately from measurement processing (in which VEE is performed and final measurements are created). Do not run these processes simultaneously.
- The duration of individual initial measurements (IMDs) within a payload should be no shorter than the time elapsed between when payloads are collected and delivered (with a minimum of 4 to 6 hours per initial measurement). For example if your system is collecting and delivering measurements three times a day, the duration of individual initial measurements should be approximately eight hours.
- If you are processing “fabricated” IMDs to migrate data, keep in mind that the system has not been tested or benchmarked for this purpose.
  For example, including 30 days’ worth of measurements in a single IMD is not typical, and has not been performance tested.

**Direct Measurement Processing Recommendations**

- If you have not enabled Initial Load IMD Ordering on your Measuring Component Types and you are leveraging the chunkSize parameter we recommend a chunkSize that will result in smaller chunks so a smaller number of IMDs are processed per chunk. This is important because the IMDs will be processed through to completion and therefore perform many more changes on the database per IMD than legacy IMD processing.

**Legacy Initial Load Measurement Processing Recommendations**

- The system includes a number of Initial Measurement Data (IMD) monitor batch processes that can be used to minimize performance issues when adding large numbers of historical or backlog readings. These use the IMD as the processing work unit instead of the Device Configuration. This prevents situations in which an error on an IMD rolls back processing for all of the IMDS for a given device configuration. These include:
  - D1-IMDV2 (IMD Monitor - Physical Devices V2): used to process initial measurements related to physical devices
  - D1-IMDD2 (IMD Monitor - Deferred IMDs V2): used to process previously held or deferred IMDs
  - D1-IMDM2 (IMD Monitor - Unattached MCs V2): used to process IMDs for standalone measuring components without a Device Configuration such as weather or price data

  Refer to the Detailed Descriptions of these batch processes for additional information about each.
- We recommend configuring a monitor process on the Pending status of the Initial Load initial measurement business objects so that the VEE and measurement creation processes can be deferred to batch processing.
  - Deferring processing has the following additional benefits:
    1. It can ensure chronological processing for scalar reads which provides more efficient processing. Processing scalar reads out of order can result in additional work as reconciliation initial measurements will be created and processed to true up the consumption calculations.
    2. It can ensure measuring components are processed in order based on their relationships. For example, if a scalar channel is configured as the check channel for an interval channel, the scalar channel will be processed first so that its data is available for sum checks and any interval gap filling required.
  - Configuring deferred processing involves adding the “IMD Monitor – Physical Devices” Batch Control as the Monitor Process on the Pending status to the following business objects:


Copyright © 2026, Oracle and/or its affiliates.

        * D1-InitialLoadIMDInterval
        * D1-InitialLoadIMDScalar
    - When initial measurement processing is deferred it will be necessary to schedule the “IMD Monitor – Physical Devices” Batch Control to run more frequently. It should be scheduled such that it runs with a higher thread count immediately after the core head end usage payload processing has completed and with fewer threads throughout the rest of day to pick up any other initial measurements that might be received (such as those received via On Demand Read commands).

**Middleware Processing Recommendations**

When processing payloads using middleware implementations, keep the number of devices to 2,000 per file for optimum usage and event processing through Oracle Service Bus. A lower number of devices, per file, will take more time for processing. A higher number of devices, per file, leads to high growth in garbage collection leading to waits and results in lower throughput. Please note, the optimum number of transactions per file may vary by head-end system.

# VEE Recommendations

- The High-Low VEE Rule is very “heavy” in terms of performance. This rule has an intensive hit on performance of the Initial Measurement loading process. Customers are encouraged to use the Usage High-Low rule, which has a less significant performance impact. If the VEE rule version is being used, the historic pre window should be set as low as possible based on business needs.
- Each VEE Rule included in VEE processing can have an impact on overall performance. The table below summaries the relative costs in terms of performance for several commonly used VEE Rules. Using this table when designing your VEE groups and rules can help identify potential performance challenges.
- For example, using "Minimal Impact" rules will likely have little to no significant impact on performance, while using several "Moderate Impact" rules could result in performance challenges, and using the High Low Check rule is very likely to introduce noticeable impacts to VEE processing performance.

| Relative Cost | VEE Rules |
|---|---|
| Minimal Impact<br>(0% - 5%) | Interval Interpolation<br>Interval Replacement Rule<br>Interval Size Validation<br>Interval Spike Check<br>Negative Consumption Check<br>Scalar Replacement Rule<br>UOM Check |
| Moderate Impact<br>(5% - 20%) | Final Measurement Validation<br>Interval Adjustment From Scalar<br>Interval Averaging<br>Multiplier Check<br>Scalar Calculation From Interval<br>Scalar Proration<br>Sum Check |
| Significant Impact<br>(20%) | High/Low Check |


Copyright © 2026, Oracle and/or its affiliates.

- Consumption sync processes should only be run on devices where the difference in consumption is outside the defined tolerance as dictated by the sum check tolerance. In a production environment, running consumption sync for two percent of devices or less is advisable.
- For cloud implementations, custom rules must be written using Groovy or Oracle Utilities Application Framework scripting.
- Run performance testing on all configuration changes performed that impact portions of Oracle Utilities Meter Data Management with high transactional processing.
- Any VEE Rules, whether provided as part of base product or custom developed, should be configured to query no more than 30 cumulative days of historical data (i.e. High/Low) to help optimize the data loading process
- A good way to troubleshoot a VEE Rule is to use the Trace section of the IMD Log. When an IMD is in Pending status, the **Trace On** button can be used to trace all VEE Rules fired during IMD processing. Once processing is complete, view the Log tab of the IMD and review the results in the **IMD Trace Log** section. For any custom developed VEE Rules, the average run time should be no longer than the average run time of other base product VEE Rules. Also, the impact of different configuration can be checked through this method as well.
- Load initial measurements and perform VEE processing every 4 to 6 hours to optimize performance.

## Measurement Processing Recommendations

- Payload processing (in which usage and event data are loaded into the system) should be run separately from measurement processing (in which VEE is performed and final measurements are created). Do not run these processes simultaneously.
- When performing Periodic Estimation for the first time, avoid running estimation for large numbers of devices for a long periods of time. The Device Periodic Estimation (D1-DVCPE) batch process used to initiate periodic estimation is designed to perform estimations for a small number of devices per run, not the entire meter population in the system.
- Set the **Perform Changes in New Session** parameter on periodic estimation algorithms (including D1-CIITBIH and D1-ARSPE) to "Y" to improve processing when many initial measurements are being created in one run of these algorithms. When this parameter is set to "Y" each initial measurement will be created in a separate session and will be committed as they are completed. This improves performance since there are fewer changes being cached and ensures progress is saved in the event of errors during processing.

## Aggregation Recommendations

- When defining the **Aggregation Horizon** for aggregation processing, the horizon should ideally be around 7 days or less (barring specific requirements that call for an longer horizon, such as in the case of some final settlement calculations).

  For each day in the horizon the system aggregates each day that many times. For example, setting the Aggregation Horizon to 30 means that each day is aggregated 30 times. The benefit of aggregating a day multiple times is that each day the quality of the data could improve if initial measurements arrive late. However there is a cost associated with a large horizon because the system is aggregating a much larger set of measurement data to catch a statistically smaller amount of changes the further into the past a day is.


Copyright © 2026, Oracle and/or its affiliates.

- A short **Aggregation Horizon** is recommended in particular when configuring Measurement Quality and Timeliness aggregations used with Oracle Utilities Analytics Visualization. Refer to Configuring Timeliness and Quality Aggregation for Use with Oracle Utilities Analytics Visualization for more information.
  The **Aggregation Horizon** is defined on Master Measurement Aggregator Type (D2-MasterMsrmtAggregatorType).
- Dynamic Aggregation is intended to combine large populations of measuring components, devices, service points, or usage subscriptions into substantially fewer aggregation measuring components. Detailed aggregation, such as at the service point or usage subscription level, is supported for small populations but does not scale to large ones, and is generally **not recommended**.

  Each foundation or composite layer creates and stores its own measuring components and measurements. Before deployment, estimate the number created at each layer; configurations approaching the source population size can duplicate large volumes of data and cause excessive scanning, processing, storage, and re-aggregation.

## Usage Transaction Recommendations

- Usage transactions should NOT be used as an extraction tool to perform the mass export of measurement data.
- Saving derived interval data and sending the data to an external system via the "Save Interval Vector" and "Extract Interval Data" options on the "Vector and Service Quantity Math" and "Get Interval Data" usage calculation rules can have a noticeable impact on usage transaction processing. Usage transaction processing can be improved by up to 45% if extraction of interval data is eliminated. Use of these options should be limited to situations in which they are absolutely required.
- Use the Usage High Low rule instead of the VEE High Low rule where possible.
- For Outbound Messages, the schema extraction for usage transactions should be converted into a RAW string operation to allow for more efficient Pre-Processing logic.

## Usage Calculation Recommendations

This section includes best practices and recommendations regarding usage calculation processing.

- Oracle recommends including the Usage Calculation High/Low Rule rule in Post-Calculation Usage Calculation Groups to validate the current usage against historical usage, and ensure that any increase or decrease of the current usage relative to historical usage is within a defined tolerance.

## User Interface Recommendations

- For better performance, user interface zones should be initially collapsed when not required for 90% or more of business processes. The initial state of zones (collapsed or not) can be controlled via the "Portal Preferences" tab on the **User** portal.
- The number of records returned to the user interface for a zone should be limited to 50 rows when building custom zones against large transactional tables.
- UI Map schemas should be specific to the displayed data. This will make sure that application will not visit all the elements that are not required for display.


Copyright © 2026, Oracle and/or its affiliates.

- A good way to troubleshoot a screen in the user interface is to go to the **Preferences** in the top right hand corner of the application and choose the **Portal Preferences** tab. Choose the appropriate portal and set all zones to "Initially Collapsed". Next, navigate back to the screen that has performance issues, expand the zones one by one, and measure the execution time of each zone. This should be an accurate step-by-step representation of the full screen execution. This testing is especially important for **custom** user interfaces.
- For optimal user interface performance, Oracle Utilities Meter Data Management users should ensure their computer set to **high performance mode** so all CPU resources are used. Also, check if allocating additional memory through your browser's default settings helps improve performance.

## Exception Handling Recommendations

- Don’t create To Do entries unless they are absolutely required
- Use the VEE Exception Handler (D1-VEERuleExceptionHandler) and Usage Exception Handler (D2-UsageRuleExceptionHandler) rules to limit the number of To Do entries created by VEE and usage rule processing.
- Do not create To Do entries for exceptions for which Service Investigative Orders are being creative.
- Purge To Do data on a regular basis (see Partitioning and Data Removal)

## SQL Recommendations

- Avoid making unbounded SQLs statements with no boundary condition on date columns.
- Reduce CLOB searches and use physical columns wherever possible.
- Implement caching to pre-fetch data instead of issuing multiple SQLs.

## Custom Rules and Processes

- When creating custom rules and processes, don’t use plug-in scripts for high volume transactions. Use Groovy or Java (on-premises implementations only).

**Cloud Services**

- Java is not allowed in cloud service environments, so implementations are encouraged to use Groovy when creating custom VEE and or usage calculation rules. Training regarding using Groovy with Oracle Utilities Application Framework applications can be found in the Oracle Utilities Learning Subscription available from Oracle University.

## Testing Recommendations

- “Day in the Life” performance testing with production-level data volumes should be performed on a Test Environment before going live.

## Java Recommendations — On-Premises Implementation Only

1. For areas where the transactional volume is high (such as Initial Measurement), use Hibernate SQL for non-CLOB fields whenever possible as opposed to reading entire


Copyright © 2026, Oracle and/or its affiliates.

Business Object. If a CLOB field must be retrieved, then either Entity or Lite BO should be used. Reading the full Business Object should be used as the last resort for high volume areas. This process should be considered for updating data as well when it provides the same functionality as updating the Business Object. These methods should only be considered if BO level validations, pre-processing, and post-processing aren't required.

2. For a simple SQL select statement needed in Java (not many joins and no complex logic), using Hibernate SQL provides a benefit over using a Business Service and a Zone since the entities are cached for Hibernate SQL.
3. "Lite" Business Objects are provided as a way to access the main fields for a BO without pulling back all of the information. Retrieving less information will speed up the process for reading the BO.

# Data Access and Retrieval Recommendations

This section includes best practices and recommendations regarding access and retrieval of data.


- Measurement Retrieval


## Measurement Retrieval

For performance reasons, measurements are written/updated directly in the database through the use of direct SQL (JDBC). Because of this, the Hibernate cache will not always be aware of changes that are made for measurements and become stale for data updated in the current session. This means that when retrieving measurements during the same session in which they were inserted into the database using the Java classes provided with the application and Hibernate can return incorrect results. We recommend using direct SQL (JDBC) to retrieve measurement data directly in these situations.

In summary:

- Pre-existing measurements that have not been altered in the current session can be retrieved using Java classes.
- Measurements that have been inserted or updated by the IMD within the current session should be retrieved using direct SQL (JDBC).

# Referencing Master Data by Identifiers


- Understanding Referencing Master Data by Identifiers


## Understanding Referencing Master Data by Identifiers

There are many places within admin configuration where direct references to master data can be made. Since master data relies on system generated keys this configuration often breaks once migrated to a new environment since the master data referenced does not exist in the target environment. To alleviate this issue, in each of these instances, a user defined identifier can be used instead of the system generated keys.

This has an added benefit for installations that support multiple time zones by enabling the identification of the master data to not only search by the user defined identifier but also the time zone of the instance data. This is advantageous because it allows one set of admin



Copyright © 2026, Oracle and/or its affiliates.

configuration to satisfy master data in multiple time zones since things like TOU Maps and profile measuring components will be identified at run time using the identifier and time zone.

The following types of data can be referenced by an identifier on admin configuration:

- **Device**: can be referenced by a device identifier type. An example of this can be seen on the Specification Lookup (D1-SpecificationLookup) extendable lookup.
- **Measuring Component**: can be referenced by a measuring component identifier. When multiple time zone support is enabled the measuring component's time zone will also be evaluated. There are two major patterns for this:
  - *Profile Factors*: The profile factor itself will reference the measuring component by identifier and then the admin configuration will point to the profile factor. An example of this is the profile factor list on interval measuring component types.
  - *Direct Measuring Component references*: An example of this is the Final Values Overlay Profile (D1-FinalValuesOverlayProfile) extendable lookup.
- **TOU Map**: can be referenced by the TOU map template code. The appropriate TOU map is then found by searching for the appropriate TOU map that references a TOU map type for the supplied TOU map template. When multiple time zone support is enabled the TOU map type time zone will also be evaluated. An example of this is the default TOU map that can be configured in the display profile of a measuring component type.

During execution the identifier supplied will be used to find the appropriate master data entry. The following validation will occur:

**When multiple time zone support is off**: the search must find one and exactly one master data entry for the identifier type and value.

**When multiple time zone support is on**: the search must find one and exactly one master data entry for the identifier type, identifier value, and time zone. Note: some installations may want to use master data across time zones, this can still be achieved by supplying one and only one master data entry for the identifier type and value:

When the search fails to find the master data for the identifier type, identifier value, and time zone and there is only one entry for the identifier type and value that master data will be used (thus allowing one master data entry to be used across time zones)

When the search fails to find the master data for the identifier type, identifier value, and time zone and there are multiple entries for the identifier type and value an error will be encountered.

# Smart Grid Gateway Native Implementations

When implementing Oracle Utilities Meter Data Management, Oracle Utilities Smart Grid Gateway, and Oracle Utilities Customer to Mater, Native implementations are recommended for both on-premises and Platform-as-a-Service (PaaS) implementations. See Smart Grid Gateway Adapter Native Implementations for more information.

# Recommendations for Creating an On-Premises Production Environment

Oracle recommends that you do not clone the demonstration environment as a basis for a new production environment. The demonstration environment typically includes transactional data that will be irrelevant to your production environment and can cause unexpected issues if it is


Copyright © 2026, Oracle and/or its affiliates.

not purged correctly. The recommended process is to start a new production environment from a new installation and migrate "clean" system data (such as business objects and algorithms) and administrative data (such as sample activity types or other administrative entities) from the demonstration and/or test or development environments as applicable.

Your implementation can use bundling and/or the Content Migration Assistant to move system and administrative data. Instructions for using these tools are contained in the Bundling and Content Migration Assistant sections in the Oracle Utilities Application Framework *Administrative User Guide*.

Contact Oracle customer support if further assistance is required.

**Recommendations for High Volume Meter Processing**

For implementations involving large volumes of meter data, Oracle recommends using Oracle Exadata. Oracle Exadata is a database machine and storage server designed to optimize enterprise-class databases and their workloads, comprising prepackaged database servers, storage servers, and storage networking.

# System-Wide Options


- Installation Options - Framework
- Feature Configurations
- Time Zones


## Installation Options - Framework


- Configuring Installation Options - Framework


### Configuring Installation Options - Framework

Installation options define the individual applications installed on your system and identify algorithms used to implement core system functions. These options also define global parameters such as the administrative menu style (alphabetical or functional), the country, language, currency code, as well as the base time zone to use for this implementation.

Installation options are stored in the installation record for your system. Use the Installation Options - Framework portal to configure these options. This portal is part of the OUAF and is described in detail in the Framework documentation.

**Base Time Zone**

The time zone setting of the Installation Options - Framework determines the time zone for all date/times stored within the system. Each date/time, based on the configuration of that field, is stored in either standard or legal time within this base time zone.

Refer to the Glossary of Terms in the *Oracle Utilities Meter Solution Business User Guide* for definitions Standard Time and Legal Time

Note: The installation record does not dictate the server time zone, but rather must match it.



Copyright © 2026, Oracle and/or its affiliates.

### Installation Algorithms

Installation algorithms implement global system functions and can be customized for each implementation. The base package supports the following installation options for Meter Data Management-related system events:

- Geocoding Service: Responsible for geocoding an address (converting an address to a geocode latitude/longitude pair).
- Global Context: Sets global contexts (displayed in the Global Context Sidebar zone) based on the value of existing global contexts. For example, if the Service Point is specified, this algorithm sets the Device by finding the most recently installed Device on the service point. It then sets the Measuring Component by finding the most effective Device Configuration and retrieving any measuring component linked to it. It then sets the Usage Subscription by finding the most recent active usage subscription linked to the service point. The contact is set by finding the main contact for the usage subscription.
  Additional detail on how global context is populated can be found in the detailed description of the D1-GBCTX-DF Algorithm Type.
- To Do Pre-creation:
  - In Meter Data Management, the D1-TDPRCRE (To Do Precreation) algorithm associates a To Do entry via characteristics to the related the related Device, Measuring Component, Service Point, Contact, Usage Subscription, Activity Type and Activity based on the drill keys of the To Do entry.
  - In Customer to Meter, the following algorithms are used to associate a To Do Entry via characteristics with related objects:
    * C1-TDCR-CHAR (Link Account/Premise/Person/SA/SP to To Do)
    * C1-TDCR-DFRL (Default To Do Role via Account/Premise)
    * D1-TDPRCRE (To Do Precreation)
    * X1-TDPRCRE (Customer to Meter To Do Pre-Creation)

> **Note**
>
> Additional detail on how the To Dos are associated to related data can be found in the detailed descriptions of these algorithms type.
>
> See Installation Options in the Oracle Utilities Application Framework *Administrative User Guide* for related information on the installation portal.

# Feature Configurations


- Configuring Feature Configurations


## Configuring Feature Configurations

Some of the features in Oracle Utilities Application Framework based applications are configured by populating options on a "feature configuration". For example, if your implementation uses Oracle Utilities Customer Care and Billing's batch scheduler, you must populate a variety of options on the batch scheduler's feature configuration.



Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to Defining Feature Configurations in the Oracle Utilities Application Framework *Administrative User Guide* for additional information.

Oracle Utilities Meter Data Management uses the following types of feature configurations (please note that more information can be found on each of these options by viewing their detailed description in the Feature Configuration portal):

### Measurement Data Options

Measurement Data Options are used to define behavior related to periodic estimation of initial measurement data, including:

- **No of Hours in Past to Retrieve Last Usable Measurement**: This option is leveraged by scalar periodic estimation to restrict how far into the past it will search for existing measurements when the last contiguous measurement is being initialized on the measuring component. This is to ensure that the first time scalar period estimation is executed on a large number of measuring components that have not been initialized for periodic estimation the system does not execute a large number of unbounded queries into the past which would result in poor performance.

### Business Intelligence Configuration

Business intelligence configuration is used to define external data source indicators used when Oracle Utilities Meter Data Management is integrated with Oracle Utilities Business Intelligence. External data source indicators allow business intelligence extracts to properly link the external identifiers to the source external system. The Value of the Data Source Indicator option should match the Environment ID on the Installation Option of the external system.

### General System Configuration

This feature configuration is owned by Oracle Utilities Application Framework but there are several important option types that have been created specific to Oracle Utilities Meter Data Management:.

- **Multi Time Zone Support**: By default the system assumes an installation operates within a single time zone. In order to enable multiple time zone functionality this option must be defined and set to "D1YS".
- **CCB Link URL**: this option is used to provide the destination URL for hyperlinking into Oracle Utilities Customer Care and Billing.
- **Trace On Flag**: can be set to automatically trace any initial measurement being processed. Note: if the initial measurement has had trace explicitly turned off then no tracing will occur. This is useful for tracing initial measurements created through automated processes where you are unable to set the trace flag directly.
- **System Override Date**: this option type is provided by OUAF but it is highly useful in testing prior to production. When set it will override the system date/time for all users. The format must be entered as: YYYY-MM-DD. For example January 1st 2010 would be 2010-01-01.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Refer to Configuring Multi-Time Zone Support for additional information on setting up multi-time zone support.

# Time Zones

- Configuring Time Zones
- Configuring Multi-time Zone Support

## Configuring Time Zones

To support businesses spanning across multiple time zones, the system stores all date and time information in a single common time zone known as the base time zone or the server time zone. Furthermore, date and time information is stored in either standard time (i.e. independent of any Daylight Savings Time adjustments, if applicable, in that time zone) or legal time (i.e. shifted according to Daylight Savings Time).

The system also allows data to be entered and displayed in a different time zone in legal time ( i.e. adjusted for Daylight Savings Time, managing the conversion back and forth between the data entry time zone and the storage time zone).

Entities associated with a physical location such as measurements (initial and final), measuring components, device configurations, devices, installation events, service points, and usage subscriptions are entered and displayed in the specific time zone where they occur, the entity time zone. The rest of the application uses the base time zone to display date and time information.

When configuring time zones the following fields are of high importance:

- **Time Zone Name**: identifies the Olson time zone and as such defines the appropriate offset from Greenwhich Mean Time as well as the schedule for shifting into and out of Daylight Savings Time.
- **Default Time Zone Label**: will be appended to date times that do not fall within Daylight Savings Time
- **Shifted Time Zone Label**: will be appended to date times that do fall within Daylight Savings Time

> **Warning**
>
> The server time zone must be correctly specified on the installation options record for the system to work properly.

For additional information see Defining Time Zones in the *Oracle Utilities Application Framework Administrative User Guide*.

## Configuring Multi-time Zone Support


Copyright © 2026, Oracle and/or its affiliates.

As a default the system assumes operations are in a single time zone. This has a few high level impacts:

- All master data must have a time zone that is equal to the base time zone. As such all master data will have the same time zone and that time zone will be the base time zone
- Certain logical complexities are avoided during high volume processing given the knowledge that all master data time zones are equal to the base time zone.

If your implementation resides within multiple time zones then the Multi Time Zone Support feature configuration must be enabled. Doing so will allow master data to be defined with any time zone configured in the system. It will also enable logic to convert time zones between the master data time zones and the base time zones.

You can access the feature configuration portal from the **Admin > General > Feature Configuration**.

From the list of results returned select the feature name for the feature type General System Configuration. For the option type Multi Time Zone Support supply "D1YS" as the value to turn multiple time zones on.

Refer to Multiple Time Zone Support for information about how functional processing is impacted by multiple time zone support.

# Defining General Options


- Units of Measure
- Service Quantity Identifiers
- Time of Use
- Service Types
- Divisions
- Entity Groups
- Weather Stations
- Postal Code Defaults
- Factors
- Markets
- Market Participants
- Understanding Processing Methods
- Processing Timetable Types


## Units of Measure


- Understanding Units of Measure
- Configuring Units of Measure


### Understanding Units of Measure



Copyright © 2026, Oracle and/or its affiliates.

Units of Measure (UOM) identify quantities measured and recorded, such as KWH, KW, cubic feet, degrees Celsius, etc. UOMs are based on a specific service type.

**Important Unit of Measure System Events**

Unit of Measure supports the following business object algorithm system events:

- **Axis Conversion**: this event receives a list of measurements along with a source UOM and interval size and a target UOM and interval size. It should then perform the necessary actions to convert the source UOM and interval size into the target UOM and interval size. Refer to the algorithm Axis Conversion algorithm (D1-AXIS-CONV) as an example.

## Configuring Units of Measure

This portal is used to display and maintain a Unit of Measure.

Refer to Understanding Units of Measure for more information.

You can access the portal by selecting **Admin**, then **General**, then **Unit Of Measure**.

The following zones may appear as part of the portal's **Main** tab page...

- **Unit of Measure List:** This zone lists all units of measure. Broadcast a record to display the details of the selected record.
- **Unit of Measure:** This zone provides information about the selected unit of measure.

# Service Quantity Identifiers

- Understanding Service Quantity Identifiers
- Configuring Service Quantity Identifiers

## Understanding Service Quantity Identifiers

Service Quantity Identifiers (SQI) are used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Some examples include: Generated, Consumed, etc.

SQIs can also be used as a stand-alone representation of a service quantity that is not measured (one that is not properly described as a UOM) within a usage service quantity collection (such as a billing determinant).

## Configuring Service Quantity Identifiers

This portal is used to display and maintain a Service Quantity Identifier.

Refer to Understanding Service Quantity Identifiers for more information.

You can access the portal by selecting **Admin**, then **General**, then S**ervice Quantity Identifier**.

The following zones may appear as part of the portal's **Main** tab page:


Copyright © 2026, Oracle and/or its affiliates.

- **Service Quantity Identifier List:** This zone lists all Service Quantity Identifiers. Broadcast a record to display the details of the selected record.
- **Service Quantity Identifier:** This zone provides information about the selected Service Quantity Identifier.

# Time of Use

- Understanding Time of Use
- Configuring Time of Use

## Understanding Time of Use

Time of Use (TOU) periods are modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.

## Configuring Time of Use

This portal is used to display and maintain a Time of Use.

Refer to Understanding Time of Use for more information.

You can access the portal by selecting **Admin**, then **General**, then **Time Of Use**.

The following zones may appear as part of the portal's **Main** tab page:

- **Time Of Use List:** This zone lists all Time Of Use records. Broadcast a record to display the details of the selected record.
- **Time Of Use:** This zone provides information about the selected Time Of Use.

# Service Types

- Understanding Service Types
- Configuring Service Types

## Understanding Service Types

Service Types define specific types of service for which usage can be recorded and captured, such as electric, gas, steam, etc.

## Configuring Service Types

This portal is used to display and maintain a Service Type.

Refer to Understanding Service Types for more information.

You can access the portal by selecting **Admin**, then **General**, then **Service Type**.


Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Type List:** This zone lists all Service Type records. Broadcast a record to display the details of the selected record.
- **Service Type:** This zone provides information about the selected Service Type.

# Divisions


- Understanding Divisions
- Configuring Divisions


## Understanding Divisions

Divisions are used to delineate between different operating companies within a large conglomerate of utilities.

Divisions can have one or more characteristics defined.

Divisions can be defined on several other objects and entities in the system, including:

- Service Point
- Service Order Activities (Request Orchestration activities)
- Service Order Field Activities
- Usage Subscription/Usage Subscription Type
- Usage Calculation Group
- Settlement Calculation Group
- Aggregation Group
- Data Source
- Market Product Type
- Market Product Set

Divisions can be associated with individual users on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

## Configuring Divisions

You use the **Divisions** portal to display and maintain divisions.

Refer to Understanding Divisions for more information.

You can access the portal by selecting **Admin**, then **General**, then **Division**.

The following zones may appear as part of the portal's **Main** tab page:

- **Division List**: This zone lists all division records. Broadcast a record to display the details of the selected record.
- **Division**: This zone provides information about the selected division.



Copyright © 2026, Oracle and/or its affiliates.

- **Characteristics**: This zone displays a list of characteristics defined for the selected division.

# Entity Groups


- Understanding Entity Groups
- Configuring Entity Groups


## Understanding Entity Groups

Entity Groups are used to define groups of entities, such as devices (including meters or items, as well as controllable devices used with Customer Program Management Cloud Service and Customer Program Management) or service points.

An entity can belong to more than one group if they have different Group Categories. For example, a controllable device associated with a specific program and rate could belong to a group category based on the specific program as well as a group category based on the specific rate.

Entity group processing uses dynamic SQL queries to identify entities that belong to a entity group. These dynamic queries are based on configuration of the Group Category (which defines Template SQL that contains tokens) and Criteria Values (which define specific values for the tokens in the Template SQL). These queries are used by "Eligible <Entity>s for this Group" zones that identify entities eligible for but not currently assigned to a group, as well as <Entity> Group Evaluation batch processes (see **Batch Processing** in **Configuring Entity Groups** for more information).

Entity Groups are defined by the following:

- **Status**: The current status of the entity group
- **Group Category**: The category to which the entity group belongs. Group categories are defined using the Entity Group Category (D1-EntityGrpCodeCategoryLookup) extendable lookup. See below for more information.
- **Group Deactivation Option**: A flag that defines what happens to the relationship (captured in the Device Entity Group or Service Point Entity Group table, as appropriate) between the group and entities belonging to the group if or when the group is deactivated. Options include "Expire" and "Do not Expire".
- **Group Assignment Option**: A flag indicating the manner in which groups are assigned to entities. Options include System (via batch process), User (manually), System and User (either).
- **Criteria Value**: One or more criteria values used for selecting entities for the group, based on a Criteria Source and Criteria Value.
  - **Criteria Source**: Defines the source for criteria used for selecting entities for the group (such as a Specification, Rate, or Program). Criteria sources are defined using the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup. See below for more information.
  - **Criteria Value**: Defines specific values for the criteria that are used when selecting entities for the group. For instance, values for a **Criteria Source** of "Rate" could be rates defined in an extendable lookup.

**System Data**



Copyright © 2026, Oracle and/or its affiliates.

Entity Groups are based on the following system data:

| Data Type | Description |
|---|---|
| Maintenance Object | Entity Group (D1-ENTGRP) |
| Business Object(s) | Entity Group (D1-EntityGroup) |

### Entity Group Category Extendable Lookup

The Entity Group Category (D1-EntityGrpCodeCategoryLookup) extendable lookup defines categories of entity groups. These are used to define groups of entities (either devices or service points) that share the criteria and template SQL for identifying and retrieving entities but that may have different criteria values. For example, two device entity groups, "Device Group A" and "Device Group B" can reference the same "Specification and Rate" category, which uses Specification and Rate Schedule criteria to identify and retrieve devices (devices in group A are those with a unique combination of Specification and Rate Schedule, while those in group B have a different unique Specification and Rate combination). The category also defines template SQL that uses the "Specification" and "Rate" criteria when identifying and retrieving devices. The Entity Group Category Template SQL contains tokens, and an Entity Groups Criteria Values define specific values for the tokens in the Template SQL.

Values for this extendable lookup are defined by the following:

- **Status**: The current status of the entity group category.
- **Target Maintenance Object**: The maintenance object of the entities that belong to entity groups of this category. Valid options include Device (D1-DEVICE) or Service Point - MDM (D1-SP).
- **Criteria Option**: Flag that indicates whether or not the criteria source along with the SQL template and SQL builder algorithm will be recorded for the entity group in this category. Valid values are "Applicable" and "Not Applicable".
- **Template SQL**: The template SQL used to identify and retrieve entities for entity groups of this category. Applicable only when Criteria Option is set to "Applicable".
- **Entity Group Algorithm**: An algorithm that executes the template SQL, inserting the criteria values from the entity group, to identify and retrieve entities for entity groups of this category.
- **Criteria Sources**: The criteria sources used by the template SQL when identifying and retrieving entities for entity groups of this category. Criteria sources are defined using the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup. See below for more information. Applicable only when Criteria Option is set to "Applicable". Each criteria source is defined by the following:
  - **Criteria Source**: The name of the criteria source (from the Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup)
  - **Is Required**: A flag that indicates if a value is required for the criteria source (Required or Optional)
  - **Marker**: The token used in the template SQL for the criteria source and value, in the format of "#xxx". For example, the marker for a Specification criteria source might be "#SPEC".
  - **Condition Operator**: The operator used in `WHERE` clauses in the template SQL to filter data based on the criteria source. Values are defined in the Condition Operator (CONDITION_OPR_FLG) lookup.

The base package includes the "SP Measurement Extract" value for this extendable lookup.


Copyright © 2026, Oracle and/or its affiliates.

### Criteria Source Extendable Lookup

The Entity Group Criteria Source (D1-EntityGrpCriteriaSrcLookup) extendable lookup defines the criteria that can be configured for an entity group. Each value defines a single criteria source, such as Specification, Rate, or Program, as well as how the user will enter criteria values for the criteria source, and the source of criteria values for the criteria source. This configuration is used in the **Criteria Value** section of an Entity Group.

Values for this extendable lookup are defined by the following:

- **Status**: The current status of the entity group criteria source
- **Presentation**: The presentation used for the criteria source. Options include:
  - **Drop down**: The user selects a the criteria value from a drop-down list.
  - **Input Text**: The user enters text.
  - **Input Text with Date Picker**: The user selects a date.
  - **Input Text with Date Time Picker**: The user selects a date and time.
- **Value Source**: The source of values for the criteria source. Options include:
  - **Characteristic Type (Pre-Defined)**: Values are from a list of pre-defined values for a characteristic type
  - **Extendable Lookup**: Values are from an extendable lookup
  - **Service Script**: Values are retrieved by a service script
  - **Standard Lookup**: Values are from a standard lookup
- **Value Retriever**: The specific value source used for the criteria source, based on the Value Source. For example, if the Value Source is "Standard Lookup", this would be the lookup code.

**Example**: A "Rate" criteria source where values are derived from the Customer Rate Schedule (D2-CCBRateScheduleLookup) extendable lookup could be defined as follows:

- **Criteria Source**: Rate
- **Description**: Rate
- **Status**: Active
- **Presentation**: Drop down
- **Value Source**: Extendable Lookup
- **Value Retriever**: D2-CCBRateScheduleLookup

## Configuring Entity Groups

You use the **Entity Group** portal to display and maintain entity groups.

Refer to Understanding Entity Groups for more information.

You can access the portal by selecting **Admin**, select **General**, then select **Entity Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Entity Group**: This zone displays basic information about entity group.


Copyright © 2026, Oracle and/or its affiliates.

- **Eligible Devices for this Group**: This zone shows a list of devices that are eligible for assignment to this entity group. The list is based on the **Template SQL** on the **Group Category** with the **Criteria Values** applied. To assign one or more devices to the group, select the devices and click **Assign**. Applicable only for "Device" entity groups.
- **Devices in this Group**: This zone shows a list of devices that have been assigned to this entity group. To delete the relationship between the device and the entity group, select the device and click **Delete**. To expire the relationship between the device and the entity group, select the device and click **Expire**. Applicable only for "Device" entity groups.
- **Eligible Service Points for this Group**: This zone shows a list of service points that are eligible for assignment to this entity group. The list is built from the **Template SQL** on the **Group Category** with the **Criteria Values** applied. To assign one or more service points to the group, select the service points and click **Assign**. Applicable only for "Service Point" entity groups.
- **Service Points in this Group**: This zone shows a list of service points that have been assigned to this entity group. To delete the relationship between the service point and the entity group, select the device and click **Delete**. To expire the relationship between the service point and the entity group, select the device and click **Expire**. Applicable only for "Service Point" entity groups.
- **Customer Address Search for Grouping**: This zone lists service points that are eligible for assignment to this entity group (and provides search parameters to filter the results). To assign one or more service points to the group, select the service points from the search results and click **Assign**. Applicable only for entity groups with a **Group Category** of "SP Measurement Extract".
- **Service Points**: This zone shows a list of service points that have been assigned to this SP Measurement Extract entity group. To unassign a service point from the entity group, select the device and click **Unassign**. To expire the relationship between the service point and the entity group, select the device and click **Expire**. To create an extract request for the group, click **Add**. Applicable only for entity groups with a **Group Category** of "SP Measurement Extract".

The following zones may appear as part of the portal's **Requests** tab page:

- **Requests**: This zone lists the requests made for the entity group, such as requests for measurement consumption extract. To a request for the entity group, click **Add** in the zone title bar.

### Batch Processing

The Device Group Evaluation (D1-DGEV) and Service Point Group Evaluation (D1-SPGEV) batch controls are used to identify devices or service points (respectively) that are eligible for a particular group and assigns them to a specified group, or expires relationships between devices or service points (respectively) and a specified group.

Devices or service points that are not currently assigned to an entity group for a specified category will be assigned to an eligible group. Only one group for the category may be assigned. If the device or service point is currently assigned a group for that category, it will be evaluated for eligibility, and if the current group is no longer eligible, the group is expired. If the same device becomes eligible for another group, that group is assigned to the device or service point.

Parameters:

- **Group Code Category**: A group category used to identify eligible entities. Required if the **Mode** is "Evaluate Group Code Category"


Copyright © 2026, Oracle and/or its affiliates.

- **Group Code**: A group used to identify eligible entities, Required if the **Mode** is "Expire Device Group Code", "Expire Service Point Group Code" or "Evaluate Group Code", and is optional if the **Mode** is "Evaluate Group Code Category".
- **Mode** (Required): The mode by which the process evaluates entities. Options include:
  - **Expire Device Group Code (D1ED)**: Used to expire device entity group relationships for the specified Group Code
  - **Expire Service Point Group Code (D1ES)**: Used to expire service point entity group relationships for the specified Group Code
  - **Evaluate Group Code (D1EG)**: Used to search entities based on Group Code, and assigns eligible devices or service points to the specified Group Code
  - **Evaluate Group Code Category (D1GC)**: Used to search entities based on Group Code Category (and optionally a Group Code). This mode assigns devices or service points eligible for a group of the specified category (and Group Code if provided), and expires relationships between devices or service points and groups of the specified category (and Group Code is provided).

Note that these batch controls are used with groups whose Group Assignment Option is either "System" or "System or User".

# Weather Stations


- Understanding Weather Stations
- Configuring Weather Stations


## Understanding Weather Stations

Weather stations are facilities that measure atmospheric conditions and collect weather data such temperature, humidity, wind speed and direction, and barometric pressure.

Weather stations are defined by the following:

- **Weather Station**; An ID for the weather station.
- **Weather Meter**: The weather meter (see below) that captures weather data for the weather station.
- **Status** A flag that indicates if the weather status is Active or Inactive.
- **Geographic Longitude**: The geographic longitude of the location of the weather station.
- **Geographic Latitude**: The geographic latitude of the location of the weather station.
- **External ID**: An ID used for the weather stations in external systems, such as a distributed energy resource management (DERM) system.

### Weather Meters

Weather meters are virtual devices used to store weather data for a weather station within the application. Weather meters can have one more interval measuring components, each configured to store a specific type of weather data (temperature, wind speed, humidity, and so on).

Weather meters are defined by the following:



Copyright © 2026, Oracle and/or its affiliates.

- **Device Type**: The device type for the weather meter. This should be based on the Weather Meter Type (D1-WeatherMeterType) business object.
- **Status**: The current status of the weather meter (Active or Inactive)
- **Serial Number**: An identifier for the weather meter. This is based on the ID of the Weather Station.

Like other devices, weather meters require device types. Device types for weather meters are created based on the Weather Meter Type (D1-WeatherMeterType) business object. In addition, weather meters have related device configurations based on the Device Configuration (D1-DeviceConfiguration) business object, and measuring components based on the Interval Channel (D1-IntervalChannel) business object.

See Creating Weather Metersfor more information about how weather meters are created.

# Configuring Weather Stations

You use the **Weather Station** portal to display and maintain weather stations.

Refer to Understanding Weather Stations for more information.

You can access the portal by selecting **Admin**, select **Geographic**, then select **Weather Station**.

The following zones may appear as part of the portal's **Main** tab page:

- **Weather Station List**: This zone lists all weather stations defined in the system. Broadcast a record to display the details of the selected record.
- **Weather Station**: This zone displays the selected weather station.
- **Weather Station Characteristics**: This zone displays characteristics for the selected weather station.

### Importing Weather Stations

Weather stations can be imported along with other facility and network location data such as transformers, feeders, and substations. Refer to Importing Facility and Network Location Data from a Distributed Energy Resource System for more information.

### Creating Weather Meters

Weather meters, including device configurations and measuring components, are created automatically when a weather station is created, based on the Device Type selected when creating the weather station.

This process requires the following data be set up prior to creating weather stations.

#### Measuring Component Type(s)

One or more measuring components used to capture weather data. Each should be defined as follows:

- **Business Object**: Interval Channel Type - Physical (D1-IntervalChannelTypePhysical)
- **Measuring Component Business Object**: Interval Channel (D1-IntervalChannel)
- **Measurement Business Object**: Measurement (D1-Measurement)
- **Service Type**: The service type associated with how weather data stored by measuring components of this type will be used (typically "Electric").


Copyright © 2026, Oracle and/or its affiliates.

- **Interval /Scalar**: Interval
- **Consumptive/Subtractive**: Consumptive
- **Interval Size**: The size of the intervals for weather data values stored by measuring components of this type, represented as hours:minutes:seconds (HH:MI:SS)
- **Value Identifiers**: One or more value identifiers for the weather data captured by measuring components of this type (Degrees F, Degrees C, and so on).

**Device Configuration Type**

A device configuration type that defines the types of measuring components to be created for the weather meter. This should be defined as follows:

- **Business Object**: Device Configuration Type (D1-DeviceConfigurationType)
- **Device Configuration Business Object**: Device Configuration (D1-DeviceConfiguration)
- **Service Type**: The service type associated with how weather data stored by devices of this device configuration type will be used (typically "Electric") This must match the service type defined for the measuring component type.
- **Valid Measuring Component Types**: One or more measuring components (defined above) used to capture weather data. Each should be marked as "Required".

**Device Type**

A device type that defines the type of weather meter to be created. This should be defined as follows:

- **Business Object**: Weather Meter Type (D1-WeatherMeterType)
- **Device Business Object**: Weather Meter (D1-WeatherMeter)
- **Service Type**: The service type associated with how weather data stored by devices of this type will be used (typically "Electric"). This must match the service type defined for the device configuration type and measuring component type.
- **Valid Device Configuration Types**: A device configuration type that defines the measuring component types to be created for the weather meter.
- **Characteristics**: Characteristics that define the type of data stored by the device (such as temperature)

### Batch Processing

The Map Service Points to corresponding Weather Stations (D1-MAPWS) batch process can be used to map service points (including metered service points) to weather stations when and as needed.

The Initial Data Export D1-WTHRSTN (D1WSTNX) batch process can be used to export weather station data when and as needed.

Use the **Batch Control** portal to view the parameters used by these batch processes.

### Importing Weather Data

Weather data can be imported into the system using the Adapter Development Kit (ADK) in the same manner as used for measurement data. When importing weather data, the weather station ID should be used as the Device ID (the Serial Number on Weather Meters is based on the Weather Station ID). In addition, this requires that appropriate Unit of Measure (UOM), Time of Use (TOU), and Service Quantity Identifiers (SQI) values are defined to represent the weather data used in your implementation, and corresponding UOM translation mapping


Copyright © 2026, Oracle and/or its affiliates.

values are defined in the ADK - UOM Code to Standard UOM Mapping (DG-HeadendUOMLookup) extendable lookup.

See About Measurements and Measurement Processing in the *Business User Guide*, and Configuring Payload Processing for more information.

# Postal Code Defaults


- Understanding Postal Code Defaults
- Configuring Postal Code Defaults


## Understanding Postal Code Defaults

Postal Code Defaults are used to define ranges of postal codes associated with a weather station. These are used when mapping service points (including metered service points) to weather stations when the longitude and latitude of the service point is not available.

Postal code defaults are defined by the following:

- **Postal Code From**: The starting point of the range of postal codes for the postal code default.
- **Postal Code To**: The ending point of the range of postal codes for the postal code default.
- **Country**: The country in which the postal code default resides.
- **City**: The city in which the postal code default resides.
- **County**: The county in which the postal code default resides.
- **State**: The state in which the postal code default resides.
- **Version**: The version of the postal code default.
- **Time Zone**: The time zone in which the postal code default resides.
- **Weather Station**: The weather station associated with the postal code default.

## Configuring Postal Code Defaults

You use the Postal Code Default portal to display and maintain postal code defaults.

Refer to Understanding Postal Code Defaults for more information.

You can access the portal by selecting **Admin**, select **Geographic**, then select **Postal Code Default - MDM**.

The following zones may appear as part of the portal's **Main** tab page:

- **Postal Code Default List**: This zone lists all postal code defaults defined in the system. Broadcast a record to display the details of the selected record.
- **Postal Code Default**: This zone displays the selected postal code default.
- **Postal Code Default Characteristics**: This zone displays characteristics for the selected postal code default.



Copyright © 2026, Oracle and/or its affiliates.

**Batch Processing**

The Data Export D1_POSTAL_DFLT (D1POSTTX) batch process can be used to export postal code default data when and as needed. Use the **Batch Control** portal to view the parameters used by this batch process.

# Factors


- Understanding Factors
- Configuring Factors


## Understanding Factors

Factor are a centrally stored set of values for use in validation rules, bill determinants calculations, and other processes.

A factor can have different values depending upon some definable attribute of a system object, such as customer size associated with a service point. Examples of factors can include minimum and maximum thresholds, loss factors, etc. Classes of factors are defined that can have numeric values (as in the above examples), or values pointing to profile measuring components, or VEE groups.

A factor's values are effective-dated values - either a number, a profile measuring component, a temperature profile measuring component, a VEE group, or some custom-defined value - assigned to a factor and associated to the value of some attribute of a system object. For example, consider a service point that can be classified as residential, commercial, or industrial. The tolerance percentage by which a customer's consumption can exceed last month's consumption can be based on the service point category.

For this example, factor values for a single factor called "tolerance percentage" could be:

- Residential - 20%
- Commercial - 10%
- Industrial - 5%.

## Configuring Factors

This portal is used to display and maintain Factors and Factor Values.

Refer to Understanding Factors for more information.

**Prerequisites**: You must define factor characteristic source algorithms, factor characteristic types, and factor characteristic values before you can create a factor. Refer to the Oracle Utilities Application Framework online help for more information about algorithms, characteristic types, and characteristic values.

You can access the portal by selecting **Admin**, then **General**, then **Factor**.

The following zones may appear as part of the portal's **Main** tab page:

- **Factor List:** This zone lists all factor records. Broadcast a record to display the details of the selected record.
- **Factor:** This zone provides information about the selected factor.



Copyright © 2026, Oracle and/or its affiliates.

- **Factor Char Value and Factor Value List:** This zone lists the characteristic values associated to the factor characteristic. For each characteristic value it will display the related factor values. From this zone you can click the **Factor Value** link to see the factor value in more detail on the **Factor Value** portal.

# Markets


- Understanding Markets
- Configuring Markets


## Understanding Markets

Markets define the jurisdictions or regulatory environments in which a service point participates.

Markets also define market relationships for valid service providers and their roles within a market (distributor, etc.). While each service point specifies only one market, a utility may serve more than one market, and different service points throughout the utility's service territory can be linked to different markets.

For each service provider defined for a market, you can also specify a fallback service provider.

### Service Providers in Deregulated Markets

Some utilities operate in deregulated markets. In implementations in deregulated markets, the system can send information to and receive information from a variety of market entities. These entities are defined as service providers.

For example, a service point's distribution company and/or energy supply company may subscribe to its consumption, or a service point's meter service provider may send requests to ping the meter that's installed at the service point to verify connectivity between the meter and its head-end system.

*Different Relationship Types in Different Markets*

Each market can define different relationship types between its service providers. A single instance of Oracle Utilities Meter Data Management or Oracle Utilities Smart Grid Gateway may have service points in different markets where each market has different relationship types and service providers. For example:

- In a regulated market the distribution company is the de facto energy supplier and meter service provider.
- Another market might have two relationship types and a single service provider for each relationship:
    1. There is a single energy supply company for the entire market
    2. There is a single meter service provider for the entire market

Yet a another market might have two relationship types (energy supply and meter service). In this market, there might be multiple service providers for each relationship type. Each service point can choose any of the relationship type's service providers. If a service point does not declare a specific service provider for a given relationship type, the relationship type's "fallback" service provider is assumed.



Copyright © 2026, Oracle and/or its affiliates.

## Configuring Markets

This portal is used to display and maintain a Market.

Refer to Understanding Markets for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market**.

The following zones may appear as part of the portal's **Main** tab page:

- **Market List:** This zone lists all Market records. Broadcast a record to display the details of the selected record.
- **Market:** This zone provides information about the selected market.

# Market Participants

- Understanding Market Participants
- Configuring Market Participants

## Understanding Market Participants

Market Participants are participants in a deregulated environment. Relationships between market participants are defined in a particular market record. Refer to Understanding Markets for more information.

Each market participant can be associated to an external system which is used to define the messages that can be sent to that market participant and how each message is sent.

## Configuring Market Participants

This portal is used to display and maintain market participants.

Refer to Understanding Market Participants for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market Participant**.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Participants List:** This zone lists all market participant records. Broadcast a record to display the details of the selected record.
- **Market Participant:** This zone provides information about the selected market participant.
- **Market Participant Characteristics**: This zone lists characteristics defined for the market participant.
- **Market Contracts**: This zone lists market contracts where the market participant is either the Buyer or Seller.
- **Processing Method List:** This zone provides the list of processing methods defined for the market participant.
- **Translation Method List:** This zone provides the list of translation methods defined for the market participant.


Copyright © 2026, Oracle and/or its affiliates.

- **Inbound BOs Send By Service Provider:** This zone lists inbound business objects that are sent by this market participant. The identification is driven by the business object having a Business Object Option of type "Sent By Service Provider" that references the current market participant.

# Understanding Processing Methods

Head end systems, external applications, and market participants can have one or more associated processing methods that define the format or means by which it receives or sends data from or to the application, such as bill determinants, interval data, or meter events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and usage transactions. Processing methods can also be used to define the information an external system wishes to subscribe to receive from our application.

At the lowest level processing methods are used to identify an outbound message type, business object, batch control, or message category and number.

Each processing method is comprised of a business object that defines what is being mapped as well as how it should be mapped.

### Important Processing Method System Events

The actual logic to determine the appropriate output for a given head end system, external application, or market participant and processing role for a processing method is executed by the following system event:

- **Determine Processing Method(s)**: is a business object algorithm system event that takes in a head end system, external application, or market participant, a processing role, and a list of relevant input data aka related object (e.g. a measuring component, device, etc). For the head end system or external application and the input data it will analyze the selection criteria to determine the appropriate output.

### How Processing Methods Work

Processing methods perform two basic tasks:

- They define the criteria for selecting the appropriate output. This can be as simple as providing a single object in return without qualification or in more involved situations it could support determining the appropriate return object based on characteristics of the data being processed. This is accomplished through the data structure defined on the processing method business object.
- They evaluate the criteria for selecting the appropriate output given a specific head end system, external application, or market participant. This is accomplished through the Determine Processing Method system event on the processing method business object.

Each processing method business object can be associated to one or more processing roles. This is done by adding the Applicable Processing Role business object option. It is these processing roles that actually create the association between a head end system, external application, or market participant, the processing method, and the functional event that is being executed. For example, when initial measurement data is processed through the IMD Seeder the processing role Initial Measurement Creation is used to identify the processing method "How to Create MC Related Information" which maps a particular measuring component type to the appropriate initial measurement business object to be used for processing.


Copyright © 2026, Oracle and/or its affiliates.

When system logic requires the results of a processing method the service Determine Service Providers and Methods is used. This service is available to call via Java or from within scripting through the business service of the same name. It can be called in one of two ways:

With a head end system, external application, or market participant: identifies the appropriate output for a single head end system, external application, or market participant being processed for the input processing role and related objects

Without a head end system, external application, or market participant: used to identify subscribing sysems. This will provide a list of any head end system, external application, or market participant that has the input processing role and an appropriate output given the related objects.

### Processing Methods Available

There are the following processing methods provided by the base package:

| Name | Details | Business Object |
|---|---|---|
| How to Create OB COMM/Send OB Message | Identifies Message Number/ Category, Business Object, Outbound Message Type and allows for an override by a device type. | D1-HowToCreateActivityOBComm |
| How to Create MC Related Information | Identifies a business object and allows for an override by measuring component type. | D1-HowToCreateMCInformation |
| How to Process Device Event Related Info | Identifies a business object, outbound message type, and batch control by device event category allowing for an override by device event type. | D1-HowToProcDvcEvtsInformation |
| How to Process Device Related Information | Identifies a business object and allows for an override by device type. | D1-HowToProcessDeviceInfo |
| How to Send Activity Related Information | Identifies a batch control and business object by activity type and allows for an override by device type. | D1-HowToSendActInformation |
| How to Send Activity Related O\B Messages | Identifies an outbound message type, message category, and message number and allows for override by activity type. | D1-HowToSendActivityResponse |
| How To Create US Related Information | Identifies a business object and allows for override by usage subscription type. | D2-HowToCreateUSInformation |
| How To Send US Related Information | Identifies a batch control, business object, and outbound message type and allows for override by usage subscription type. | D2-HowToSendUSInformation |
| How to Process Service Point Related Info | Identifies a business object and allows for override by service point type. | D1-HowToProcSPRelatedInfo |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| How to Send Field Activity Related Info | Identifies a outbound message type and allows for an override by field task type. | D1-HowToSendFARelatedInfo |
| How to Send Field Activity Remark Info | Identifies an outbound message type for an activity remark type. | D1-HowToSendActivityRemarkInfo |
| How to Translate External Value | Identifies a business object and allows for override by identifier. | D1-HowToTranslateExternalValue |
| How to Request Customer Notification | Identifies a list of outbound message types. | D1-HowToRequestCustomerNotific |
| How to Process Business Flag Related Info | Identifies an outbound message type and allows for an override by business flag type. | D1-HowToProcessBusinessFlagInf |

# Processing Timetable Types


- Understanding Processing Timetable Types
- Configuring Processing Timetable Types


## Understanding Processing Timetable Types

Processing timetable types define types of schedules that can be referenced by different processes and objects in Oracle Utilities applications such as Oracle Utilities Meter Data Management and Oracle Utilities Market Settlements Management. For example, if a process requires requesting data from an external system on a daily basis, a daily processing timetable type could be used to define the details of this schedule.

Processing timetable types are used by standalone data request measuring components to control the schedule by which requests for data are processed (for example, in a settlement process in which zonal load is requested on a daily basis). They are also used in the attribute data snapshot creation process.

The base package includes two processing timetable type business objects used with specific scheduling patterns:

- **Daily Processing Timetable**: Used for daily schedules
- **Defined Dates Processing Schedule**: Used for schedules that should execute on specific dates.

Parameters used to define processing timetable types include:

- **Execution Frequency Class**: Designates the type of processing timetable, Daily or Defined Dates (based on the business object)
- **Scheduling Information**: Details about how the schedule is executed, including:
  - **Period Cut-Off Time**: The time of day that represents the end of a period. In many instances this will be 12:00 AM. However, for something like a "Gas Day" this might be set to 09:00 AM.
  - **Earliest Execution Time**: Used to prevent records from being processed until after this time each day. This time relates to the processing time.



Copyright © 2026, Oracle and/or its affiliates.

  - **Lag / Lead**: A flag that indicates whether periods in the past (Lag) or future (Lead) should be processed (applies to Daily schedules only)
  - **Lag Days / Lead Days**: The number of days in the past (Lag) or future (Lead) that should be processed (applies to Daily schedules only)
  - **Restrict Processing by Work Calendar**: Used to restrict processing to only the work days configured in the Work Calendar and will exclude any holidays added to the calendar (applies to Daily schedules only)
- **Workflow Information**: Details about how the processing timetable aligns with process workflows, including warning time and target due time
- **Processing Dates**: A list of specific dates on which the process should be executed (used with Defined Dates Processing Schedule only)

## Configuring Processing Timetable Types

Refer to Understanding Processing Timetable Types for more information.

You can access the portal by selecting **Admin**, then **General**, then **Processing Timetable Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Processing Timetable Type List**: This zone lists all processing timetable type records. Broadcast a record to display the details of the selected record.
- **Processing Timetable Type**: This zone displays details for the selected processing timetable type.

# Defining Asset Options

> **Note**
>
> The asset management functionality described in this section is available with the following products:
>
> - Oracle Utilities Advanced Meter Solution (C2M-AMS)
> - Oracle Utilities Customer to Meter (C2M)
> - Oracle Utilities Customer Cloud Service (CCS)
> - Oracle Utilities Meter Data Management (MDM)
> - Oracle Utilities Meter Solution Cloud Service (MSCS)

- Asset Activity Types
- Asset Completion Event Types
- Asset Inbound Communication Types
- Asset Types
- Service History Questions
- Service History Types


Copyright © 2026, Oracle and/or its affiliates.

- Configuration Types
- Configurations
- Configuration Reports

# Asset Activity Types

- Understanding Asset Activity Types
- Defining Asset Activity Types

## Understanding Asset Activity Types

Activity types define the main attributes broad range of activities, such as maintenance, repairs, inspections, field work etc. The base package includes an asset activity type business object (W1-SOMActivityType) that should be used when created asset activity types.

Activity types are defined by:

- **Activity Business Object**: The business object used for asset activities of this type
- **Activity Schedule Information**: Default scheduling information for asset activities of this type, including Activation Lead Days, Work Lead Days, and Activity Disposition.
- **Attachments**: One or more attachments associated with asset activities of this type, such as procedures or checklists.
- **Exception Handling**: Details regarding exception handling for asset activities of this type, including default To Do Type, To Do Role, and retry options.
- **Completion Event Types**: One or more completion event types associated with asset activities of this type. See Understanding Completion Event Types for more information.
- **Object To Create**: The service history type to be created by asset activities of this type. Service history records are created by Create Activity Type's Service History completion events.

Refer to Defining Asset Activity Types for more information about setting up asset types.

> **Note**
>
> Asset activity types are created by the system when field task types are created in Service Order Management.

## Defining Asset Activity Types

You use the **Activity Type** portal to display and maintain asset activity types.

Refer to Understanding Asset Activity Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Activity Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:


Copyright © 2026, Oracle and/or its affiliates.

- **Activity Type**: This zone displays basic information about the activity type.
- **Attachments**: This zone attachments associated with the activity type.

# Asset Completion Event Types


- Understanding Asset Completion Event Types
- Defining Asset Completion Event Types


## Understanding Asset Completion Event Types

Asset completion event types are used to define various processes or events to be executed when completing an asset activity such as asset replication, service history creation, attachment creation, and activity creation.

Asset completion event types are referenced on activity types to indicate which completion events should be included as part of the lifecycle of activities of that type. For example, this affects when service history is created for an activity, and the disposition of the assets on an activity as the activity is processed. Asset completion events can be configured as optional or required for activities of different types.

Asset completion event types are defined by:

- **Completion Event Business Object**: The business object used for asset completion events of this type
- **Identifiers**: One or more identifiers that can be used to characterize asset completion events of this type
- **Inbound Communication**: Information used to create asset completion events of this type using information defined within an inbound communication business object, or within its raw message
- **UI Payload Create Data Mover**: XPath statements executed when creating asset completion events of this type online. These statements are used to populate elements within the payload from the completion event business object.

Refer to Defining Completion Event Types for more information about setting up completion event types.

## Defining Asset Completion Event Types

You use the **Completion Event Type** portal to display and maintain asset completion event types.

Refer to Understanding Asset Completion Event Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Completion Event Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Completion Event Type** : This zone provides information about the selected completion event type, including as inbound communications and attachments.



Copyright © 2026, Oracle and/or its affiliates.

- **Unlink Activity Types**: This zone enables the removal of an activity type from the completion event type. To remove an activity type select the activity type(s) to remove and click **Remove**. The selected activity types will no longer appear in the **Unlink Activity Types** zone.
- **Link Activity Types**: This zone enables adding an activity type to the completion event type. To link an activity type:
    1. Enter search criteria n the **Activity Type Description** field and click **Search**. The zone will display a list of activity types matching your search criteria.
    2. Select the activity type(s) to add and click **Link**. The selected activity types will now appear in the **Unlink Activity Types** zone.

# Asset Inbound Communication Types


- Understanding Asset Inbound Communication Types
- Defining Asset Inbound Communication Types


## Understanding Asset Inbound Communication Types

Asset inbound communication types define the information required for the system to receive messages from external systems. These typically hold value mappings of the inbound communication elements to an xpath element in the completion message.

Asset inbound communication types are defined by:

- **Inbound Communication Business Object**: The business object used for asset inbound communications of this type
- **Exception Handling**: Details regarding exception handling for asset inbound communications of this type, including default To Do Type, To Do Role, and retry options.
- **Message Validation**: Source elements within the inbound communication business object or raw message used for validation of asset inbound communications of this type
- **Communication Data Mapping**: Information used to populate communication business object elements from information within the communication's raw message
- **Attachment Suppression**: Substitution text used in place of large data attachments in the completion event list on the inbound communication. Used when messages received from an external system contain attachment data that are very large.

Refer to Defining Asset Inbound Communication Types for more information about setting up asset inbound communication types.

## Defining Asset Inbound Communication Types

You use the **Communication Type** portal to display and maintain asset inbound communication types.

Refer to Understanding Asset Inbound Communication Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Inbound Communication Type**.

The following zones may appear as part of the portal's **Main** tab page:



Copyright © 2026, Oracle and/or its affiliates.

- **Communication Type List**: This zone lists all asset inbound communication type records. Broadcast a record to display the details of the selected record.
- **Communication Type**: This zone provides information about the selected communication type, such as attachment suppression, communication data mapping, message validation and exception handling.

# Asset Types


- Understanding Asset Types
- Defining Asset Types


## Understanding Asset Types

Asset types define the attributes for assets and components of a certain type including information such as the valid location types and service history types for the assets and whether or not such assets can have attached components.

Asset types are defined by:

- **Component Relationship**: A flag that indicates if components can be installed on assets or components of this type
- **Asset Business Object**: The business object used for assets or components based on this asset type
- **Valid Location Types**: A list of location types where assets or components of this type can be located
- **Valid Attached To Asset Types**: A list of valid asset types where components of this type can be attached (applicable to components only)
- **Valid Service History Types**: A list of valid service history types that can apply for assets or components of this type

Refer to Defining Asset Types for more information about setting up asset types.

### Asset Classes

Assets are defined by classes, each of which is based on a specific business object, and determines the type of asset: an asset or a component.

The table below lists the asset classes and the business object for each.

| Asset Class | Business Object |
|---|---|
| Asset | W1-AssetType (Operational Device Asset Type) |
| Component | W1-ComponentType (Operational Device Component Type) |

## Defining Asset Types

You use the **Asset Type** portal to display and maintain asset types.

Refer to Understanding Asset Types for more information.



Copyright © 2026, Oracle and/or its affiliates.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Asset Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset Type** : This zone provides information about the asset type.
- **Asset Type Statistics**: This zone displays statistics for the asset type.

# Service History Questions


- Understanding Service History Questions
- Defining Service History Questions


## Understanding Service History Questions

Service History Questions are used to guide workers through corrective and preventive inspection processes as part of an asset's service history.

Service history questions are associated to a service history type which is then associated to an asset type.

During a corrective or inspection process, workers can be guided through a series of questions assessing the condition of the asset. The answers can be "weighted", which when calculated, gives an overall score as to the condition of the asset. The calculation is generally a sum (answer points * question weight), but can vary depending on the service history type. The total inspection points are then normalized to an overall grade or score. The condition rating and confidence rating are also calculated and the asset is updated with the appropriate values.

A service history question's answers can be defined as check boxes, multiple choice, or free-form text.

Refer to Defining Service History Questions for more information about defining service history questions.

## Defining Service History Questions

You use the **Service History Question** portal to display and maintain Service History questions.

Refer to Understanding Service History Questions for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Service History Question**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Question**: This zone provides information about the question.
- **Service History List**: This zone displays a list of service history types that reference the question.



Copyright © 2026, Oracle and/or its affiliates.

# Service History Types


- Understanding Service History Types
- Defining Service History Types


## Understanding Service History Types

Service history types define the main attributes for service history of a certain type. Types might include maintenance, inspection, service, test, measurement, and so on.

Service history types are defined by the following:

- **Service History Business Object**: The business object used to create service history records of this type.
- **Category**: The category to which service history records of this type belong. The base package includes the following categories:
  - Downtime
  - Failure
  - Inspection
  - Maintenance
  - Repair
- **Subclass**: The specific subclass of the service history type, based on the **Service History Business Object**. Oracle Utilities Customer To Meter supports two subclasses of service history types:
  - Operational Device Service History
  - Questionnaire Service History

  Each service history type subclass includes other specific configuration information, outlined below.

#### Operational Device Service History Types

Operational Device Service history types are also defined by the following:

- **Service Schedules**: One or more template work orders and time intervals that should be used to schedule work order generation for service history records of this type

#### Questionnaire Service History Types

Questionnaire Service history types are also defined by the following:

- **Follow-up Actions**: One or more actions intended as “follow-up” actions for service histories of the service history type, such as work orders or work order activities, users can configure those follow-up actions on the service history type. Note that if a regulatory compliance requirement is associated with a follow-up action, a compliance type must be specified. For example, a Service History Type of “Grade 2 Gas Leak”, might include a follow-up action of “Recheck” which could be configured with a “30 day” Compliance Type and a follow-up action of “Repair” which could be configured with a “1 year” Compliance Type.



Copyright © 2026, Oracle and/or its affiliates.

- **Stop Processing Service History Type**: One or more service history types that should stop the processing of **Follow-ip Actions**. For example, certain problem that needs follow up actions was reported. Then another service history, which indicates the problem has been fixed or its severity has changed, is recorded. Businesses might want to stop/cancel the follow up actions based on the original problem/severity.
- **Questions**: A list of one or more questions used to guide inspection of an asset. Questions are defined in the **Questions** portal.

## Defining Service History Types

You use the **Service History Type** portal to display and maintain service history types.

Refer to Understanding Service History Types for more information.

You can access the portal by selecting **Admin**, then **Asset Management**, then **Service History Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Service History Type**: This zone provides information about the service history type.
- **Questionnaire Tree**: This zone displays the service history types' questionnaire, based on the questions defined for the service history type. Applicable for Questionnaire service history types only.
- **Unlink Asset Types**: This zone lists the asset types currently linked to the service history type. To unlink one or more asset types, check the corresponding checkboxes and click **Remove**.
- **Link Asset Types**: This zone is used to search for and link asset types to the service history type. To link new asset types,
  1. Enter search criteria in the **Asset Type Description** field.
  2. Click **Search**.
  3. Select the asset types to link by checking the corresponding checkboxes.
  4. Click **Link**.

# Configuration Types


- Understanding Configuration Types
- Defining Configuration Types


## Understanding Configuration Types

Configuration types define the information required for various asset configurations with other assets, components, firmware, and other data.

Configurations are referenced on specifications which are then referenced on assets and components to indicate the best way that these entities can work with the other entities that they are connected to in the field.



Copyright © 2026, Oracle and/or its affiliates.

Configuration types are defined by the following:

- **Configuration Class**: The class for configurations of this type
- **Configuration Category**: The category of configurations of this type
- **Configuration Business Object**: The business object used for configurations of this type

### Configuration Classes

In general configuration classes define the types of connections that are being configured. Each configuration class is based on a specific business object, which is the business object used for configurations of each class and type.

The table below lists the configuration classes and the business object for each.

| Configuration Class | Business Object |
|---|---|
| Single Asset | W1-SingleAssetConfig (Single Asset Configuration) |
| Asset/Component | W1-AssetCompConfig (Asset/Component Configuration) |
| Asset to Asset | W1-AssetToAssetCompConfig (Asset to Asset/Component Configuration) |
| Asset/Component to Asset | W1-AssetCompToAssetConfig (Asset/Component to Asset Configuration) |
| Asset to Asset/Component | W1-AssetToAssetConfig (Asset to Asset Configuration) |
| Asset/Component to Asset/ Component | W1-AssetCompToAssetCompConfig (Asset/Component to Asset/ Component Config) |

## Defining Configuration Types

You use the **Configuration Type** portal to display and maintain configuration types.

Refer to Understanding Configuration Types for more information.

You can access the portal by selecting **Admin**,, then **Asset Management**, then **Configuration Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Configuration Type List**: This zone lists all configuration type records. Broadcast a record to display the details of the selected record.
- **Configuration Type** : This zone provides information about the selected configuration type.

# Configurations

- Understanding Configurations
- Defining Configurations

## Understanding Configurations


Copyright © 2026, Oracle and/or its affiliates.

Configurations document and manage valid setups for assets and components. This includes a specification for each asset or component in the configuration as well as the valid value for each parameter, such as the firmware version of each asset.

Configurations are defined by the following:

- **Configuration Type**: The configuration type upon which the configuration is based
- **Status**: The current status of the configuration
- **Configuration Detail**: Specific information related to the configuration such as the assets and components involved, the firmware, characteristics, identifiers and other important information

## Defining Configurations

You use the **Configuration** portal to display and maintain configurations.

Refer to Understanding Configurations for more information.

You can access the portal by selecting **Admin**, then **Configuration Report**, then **Configuration**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Configuration**: This zone provides information about the selected configuration.
- **Configuration Attachments**: This zone lists any attachments defined for the configuration.

# Configuration Reports

- Understanding Configuration Reports
- Defining Configuration Reports

## Understanding Configuration Reports

Configuration reports provide a summary of any currently installed configurations.

These reports show whether the configuration in an installation is valid or invalid, the identifiers and values being used, and display all of the assets/components and other elements that make up the installed configuration.

Configuration reports are defined by the following:

- **Reporting Mode**: The mode in which the report is run (One Time or Recurring)
- **Status**: The current status of the report (Pending or Submitted)
- **Report Criteria**: Specific information used to establish the filter criteria when generating the report, such as the assets and components involved, the firmware, characteristics, identifiers and other important information.
- **Report Result**: Specific information for each configuration, including the number of configurations, the assets and components involved, and the specifications and firmware


Copyright © 2026, Oracle and/or its affiliates.

versions of each. If the configuration is invalid, the **Configuration** column is blank in the report results. Users can navigate to specific valid configurations to review and edit the setup as needed. In addition, users can generate a work order to correct, reconfigure, or otherwise maintain any of the configurations by clicking **Generate Activity**.

### Base Package Configuration Reports

The base package includes a number of configuration reports, each based on one of the support configuration classes (see Understanding Configuration Types.

The table below lists the configuration reports and the business object for each.

| Configuration Report | Business Object |
|---|---|
| Single Asset Configuration Report | W1-SingleAssetCfgRpt |
| Asset/Component Configuration Report | W1-AssetCompCfgRpt |
| Asset to Asset Configuration Report | W1-AssetToAssetCfgRpt |
| Asset/Component to Asset Configuration Report | W1-AssetCompToAssetCfgRpt |
| Asset to Asset/Component Configuration Report | W1-AssetToAssetCompCfgRpt |
| Asset/Component to Asset/Component Configuration Report | W1-AssetCompToAssetCompCfgRpt |

## Defining Configuration Reports

You use the **Configuration Report** portal to display and maintain configuration reports.

Refer to Understanding Configuration Reports for more information.

You can access the portal by selecting **Admin**, then **Configuration Report**, then **Configuration Report**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Configuration Report**: This zone provides information about the selected configuration report.

# Defining Location Options

> **Note**
>
> The asset location functionality described in this section is available only to cloud implementations using Oracle Utilities Meter Solution Cloud Service.

- Asset Location Types
- Organization Types
- Organizations
- Out of Service Location Types


Copyright © 2026, Oracle and/or its affiliates.

- Out of Service Locations

# Asset Location Types

- Understanding Asset Location Types
- Defining Asset Location Types

## Understanding Asset Location Types

Asset location types define the details of locations of a certain type where assets or components can be installed. Valid types of asset locations can include service points, other locations for installed assets, or any other location as defined by your organization.

Asset location types are defined by the following:

- **Class**: The general class for asset locations of this type, such as commercial or residential, a shop or warehouse, etc.
- **Location Business Object**: The business object used for asset locations of this type
- **Number of Assets Allowed**: A flag that indicates whether or not asset locations of this type can accommodate a single or multiple assets
- **Service Type**: The type of service (electric, water, etc.) that asset locations of this type support
- **Allow Related Work Locations**: A flag that indicates whether asset locations of this type should be allowed to be connected to a work location
- **Valid Parent Location/Organization Types**: One or more valid parent locations and parent organizations that can be referenced on asset locations of this type. Used primarily for reporting.
- **Valid Communicate-To Location Types**: One or more asset location types to which asset locations of this type can communicate. The **Tree View** tab of any asset or location will display other related assets with a visual indication of their relationship.

> **Note**
>
> Asset location types are used to define locations where assets and components are installed. Locations where assets that are out of service are stored and maintained are defined via Out of Service locations and Out of Service location types.

## Defining Asset Location Types

You use the **Asset Location Type** portal is used to display and maintain asset location types.

Refer to Understanding Asset Location Types for more information.

You can access the portal by selecting **Admin**, **Location**, then **Asset Location Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Asset Location Type List**: This zone lists all asset location type records. Broadcast a record to display the details of the selected record.


Copyright © 2026, Oracle and/or its affiliates.

- **Asset Location Type**: This zone provides information about the selected asset location type.

# Organization Types

- Understanding Organization Types
- Defining Organization Types

## Understanding Organization Types

Organization types define the main attributes for organizations of a certain type.

Organization types are defined by the following:

- **Class**: The general class organization. Parent organizations have a class of “Parent”.
- **Organization Business Object**: The business object used for organizations of this type
- **Valid Parent Organization Types**: One or more valid parent organizations that can be referenced on organizations of this type. Used primarily for reporting.

## Defining Organization Types

You use the **Organization Type** portal is used to display and maintain organization types.

Refer to Understanding Organization Types for more information.

You can access the portal by selecting **Admin**, then **Location**, then **Organization Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Organization Type List**: This zone lists all organization type records. Broadcast a record to display the details of the selected record.
- **Organization Type**: This zone provides information about the selected organization type.

# Organizations

- Understanding Organizations
- Defining Organizations

## Understanding Organizations

Organizations are used for categorization and reporting for locations.

For example, your reporting may include a hierarchy that groups locations by a structure of parent and child organizations.

Organizations are defined by the following:

- **Organization Type**: The organization’s type. Understanding Organization Types for more information.


Copyright © 2026, Oracle and/or its affiliates.

- **Organization Disposition**: The current status of the organization (Active or Inactive)
- **Parent Organization**: The parent organization, if applicable.
- **Main Contact**: A contact for the organization

## Defining Organizations

You use the **Organization** portal to display and maintain organizations.

Refer to Understanding Organizations for more information.

You can access the portal by selecting **Admin**, then **Location**, then **Organization**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Organization**: This zone provides information about the organization.
- **Child Locations / Organizations**: This zone lists any child locations and/or organizations for the current organization.

# Out of Service Location Types


- Understanding Out of Service Location Types
- Defining Out of Service Location Types


## Understanding Out of Service Location Types

Out of service location types define attributes of types of locations where assets that are out of service are stored and maintained. Examples of out of service locations include warehouses, storerooms, and repair shops.

Out of service location types are defined by the following:

- **Class**: The general class (storeroom, meter shop, etc.) of out of service locations of this type
- **Location Business Object**: The business object used for out of service locations of this type
- **Valid Parent Location/Organization Types**: One or more valid parent locations and parent organizations that can be referenced on out of service locations of this type. Used primarily for reporting.

## Defining Out of Service Location Types

You use the **Out of Service Type** portal is used to display and maintain out of service location types.

Refer to Understanding Out of Service Location Types for more information.

You can access the portal by selecting **Admin**, then **Location**, then **Out of Service Location Type**.



Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Out of Service Location Type List**: This zone lists all out of service location type records. Broadcast a record to display the details of the selected record.
- **Out of Service Location Type**: This zone provides information about the selected out of service location type.

# Out of Service Locations


- Understanding Out of Service Locations
- Defining Out of Service Locations


## Understanding Out of Service Locations

Out of service locations are physical locations where assets are stored or repaired such as a storeroom, warehouse, or service facility. When assets are changed to statuses such as “In Repair” or “Pending Disposition” the asset location is typically moved to an out of service location.

Out of service locations are defined by the following

- **Location Type**: The out of service location’s type. See Understanding Out of Service Location Types for more information.
- **Location Disposition**: The current status of the out of service location (Active or Inactive)
- **Parent Location/Organization**: The out of service locations’ parent location or organization (if applicable)
- **External ID**: An ID used to identify the out of service location in external systems
- **Address Information**: The physical address of the out of service location
- **Contacts**: The primary contact for the out of service location. See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with out of service locations.
- **Field Information**: Additional information about the physical location, such as GPS coordinates

## Defining Out of Service Locations

You use the **Out of Service Location** portal to display and maintain out of service locations.

Refer to Understanding Out of Service Locations for more information.

You can access the portal by selecting **Admin**, then **Location**, then **Out of Service Location**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Out of Service Location**: This zone provides information about the out of service location.
- **Current Activities**: This zone provides a summary of the activities that are currently associated to the location.



Copyright © 2026, Oracle and/or its affiliates.

- **Asset Summary**: This zone provided statistics about the number of assets, by asset type, that are currently associated with the location.
- **Owned Attachments**: This zone lists attachments owned by the location.
- **Referenced Attachments**: This zone lists attachments related to the location.

> **Note**
>
> See About Contacts and Maintaining Contacts for more information about creating and maintaining contacts used with out of service locations.

# Defining Device Options


- Command Sets
- Manufacturers
- Head End Systems
- Measuring Component Types
- Measuring Component Comparison Types
- Device Configuration Types
- Device Types


## Command Sets


- Understanding Command Sets
- Configuring Command Sets


### Understanding Command Sets

Command sets are used to define a group of commands that are not eligible for a particular device. For instance, if Commission or Decommission commands should be considered ineligible for a particular device model, a command set that references the Device Commission and Device Commission business objects could be created and associated with that device model.

Command sets are specified for individual device models via the Manufacturer portal.

Individual devices of a particular model can be configured to override ineligibility if needed.

### Configuring Command Sets

This portal is used to display and maintain a Command Set.

Refer to Understanding Command Sets for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Command Set**.



Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Command Set List:** This zone lists all Command Set records. Broadcast a record to display the details of the selected record.
- **Command Set:** This zone provides information about the selected Command Set.

# Manufacturers


- Understanding Manufacturers
- Configuring Manufacturers


## Understanding Manufacturers

Manufacturers are the companies that makes devices.

A device's manufacturer is defined as an attribute of the device itself.

Each manufacturer can have zero or more models defined. Models for a single manufacturer can have diverse service types.

- Models can specify an **Exclude Command Set** that references commands that are not eligible for that model. Refer to Understanding Command Sets for more information.
- The **Device Command Set Override** field indicates if a command set defined by in **Exclude Command Set** field may be can be overridden and specified at the device.

## Configuring Manufacturers

This portal is used to display and maintain a Manufacturers.

Refer to Understanding Manufacturers for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Manufacturer**.

The following zones may appear as part of the portal's **Main** tab page:

- **Manufacturer List:** This zone lists all Manufacturer records. Broadcast a record to display the details of the selected record. **Manufacturer:** This zone provides information about the selected Manufacturer.

# Head End Systems


- Understanding Head End Systems
- Understanding SGG Adapter Configuration
- Configuring Head End Systems


## Understanding Head End Systems

Head end systems are systems that collect measurement data and meter events for eventual submission to the application. Many devices can communicate to the application through a



Copyright © 2026, Oracle and/or its affiliates.

single head-end system, but a utility may have numerous head-end systems through which they communicate with devices.

Head end systems utilize processing methods that specify the type of initial measurement data and device events to create for devices (and their related measuring components) based on measuring component type. Head end systems also utilize processing methods that specify how smart meter commands are processed.

Refer to Understanding Processing Methods for more information about processing methods.

**Head End Systems Impact Data Import and Export**

Head end systems are configured to identify how a particular external system communicates data with Oracle Utilities Meter Data Management. This includes:

- The identifier type used to locate devices and measuring components. These are used both on import and export of data.
- The date/time format used in various data imports (i.e. whether or not the date/time format includes time zone information).

Please refer to the embedded help for more information about these fields.

Each head end system can be associated to an external system which is used to define the messages that can be sent to that service provider and how each message is sent.

# Understanding SGG Adapter Configuration

This section describes how to use a head end system's SGG Adapter Configuration portal.

You can use the SGG Adapter Configuration portal to view configuration information and access configuration components for an SGG adapter head end system.

Note: This portal displays configuration information for head-end systems that reference an SGG Adapter Configuration Sheet extendable lookup..

To use the configuration information portal for an SGG adapter head end system:

Select **Admin > Device > Head End System**.

In the Head End system List zone, click the Broadcast icon for the head-end system you wish to view.

Click the SGG Adapter Configuration tab to view the configuration information.

The SGG Adapter Configuration portal contains the following zones:

- **SGG Adapter Configuration Tracker**: This zone displays the configuration details of the adapter, as defined by the SGG Adapter Configuration Sheet extendable lookup referenced on the head end system. The configuration details include:

The components required for usage and event processing and command processing. To view more details about the components, you can click the component name to go to the business object for the component. For example, you can click the business object "SSN - Connect or Disconnect" to go to the business object portal for the SSN - Connect or Disconnect business object.

Status messages describing the configuration status of components. The following table lists the status messages that may be displayed and the possible actions you can take:

| If the status message is... | You can... |
|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | |
|---|---|
| Set up this processing method | Click the processing role to set up the processing method. |
| This processing method has been configured | Click the processing role to view the configured processing method. |
| Update your processing method with a communication BO | Click the status message to set up the processing method. |
| Update your external system / outbound message type with an Message sender | Click the status message to go to the external system. |
| Update your processing method with an outbound message type | Click the status message to set up the processing method. |
| Add a value to get started | Click the status message to go to the extendable lookup. |
| Values Existing: (number) | Click the status message to go to the extendable lookup. |
| Add a communication type | Click the status message to go to the communication type portal. |
| Add a device event type | Click the status message to go to the device event type business object for the communication type. This message appears only for Echelon type adapters. |
| Master Configuration has been added | Click the status message to view the Master Configuration portal. |
| Add Master Configuration for this adapter | Click the status message to view the Master Configuration portal. |

**Upload Statistics Aggregators**: This zone lists the IMD Upload Statistics Aggregator measuring components associated with the head-end system.

# Configuring Head End Systems

This portal is used to display and maintain Head End Systems.

Refer to Understanding Head-End Systems for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Head End System**.

The following zones may appear as part of the portal's **Main** tab page:

- **Head End System List:** This zone lists all Head End System records. Broadcast a record to display the details of the selected record.
- **Head End System:** This zone provides information about the selected Head End System.
- **Processing Method List:** This zone provides the list of processing methods defined for the Head End System.
- **Translation Methods List:** This zone provides the list of translation methods defined for the Head End System.
- **Inbound BOs Send By Service Provider:** This zone lists inbound Business Objects that are sent by this Head End System. The identification is driven by the Business Object having a Business Object Option of type "Sent By Service Provider" that references the current Head End System.


Copyright © 2026, Oracle and/or its affiliates.

# Measuring Component Types


- Understanding Measuring Component Types
- Configuring Measuring Component Types


## Understanding Measuring Component Types

Measuring component types define the most important properties of a measuring component.

Measuring component types define what a measuring component measures (KWH, temperature, etc.), how regularly it measures it, and whether it should be connected to a physical device, or if it's used as a scratchpad measuring component or an aggregator measuring component. Measuring component types also specify how the measuring component's final measurements should be stored, how the measuring component's user-defined values should be calculated, and specific rules governing validation, editing, and estimation (VEE) for measuring components of the type. In addition, measuring component types define display properties and valid attribute values for measuring components belonging to the type.

The following configurable items are available for most measuring component types:

- **Anchor Time**: A time used to define the start and end time of the business day for purposes of processing measurement data and usage transactions for measuring components of this type. Applicable only to dynamic aggregation measuring component types, standalone measuring component types, and profile measuring component types.
- **Initial Load IMD Ordering (Fallback)**: A flag that indicates if incoming initial measurements for measuring components of this type should be processed using Measuring Component Ordering. This is used most often in the case of daily scalar measuring components for which data is occasionally sent to the system out of order, but can also be used to ensure that "check channels" are processed before related channels. Applicable only to measuring component types used with Direct Measurement Processing (see Direct Measurement Processing and Measuring Component Ordering for more information).
- **Value Identifiers**: These store the values of UOM, TOU, and SQI that identify the measured amounts for measuring components of this type. Value identifiers specify the quantities stored on the measurement records for measuring components of this type. Please refer to the Measuring Component Type Value Identifiers topic later in this section for more information.
- **Valid VEE Groups**: These define the VEE groups considered valid for measuring components of this type. Each group supplied here will be available to be selected on measuring components of this type and act as an override to the Fallback VEE Groups.
- **Fallback VEE Groups**: These define default VEE groups for a specific VEE Group Role that can be used with all measuring components of this type. This alleviates the need to specify the same VEE groups on multiple measuring components of the same type. Changes made to these groups will automatically apply to all measuring components of this type unless they have specified their own VEE groups for that particular VEE Group Role. Each VEE group is designated a VEE group role that indicates when and how the VEE group is used (for initial load, manual override, estimation, etc.).
- **Enable ML Based Validation (Fallback)**: A flag that indicate (Yes or No) if initial measurements for measuring components of this type will be processed through a machine learning scoring model. IF set to "Yes", VEE processing for initial load data will



Copyright © 2026, Oracle and/or its affiliates.

only be used when data fails machine learning normalcy checks. See Machine Learning Anomaly Scoring for more information.

- **Anomaly Score Threshold (Fallback)**: The anomaly score threshold for measuring components of this type. This overrides the value set in the MDM Master Configuration. This field is only visible when the **Enable ML based Validation (Fallback)** flag is set to "Yes". See Machine Learning Anomaly Scoring for more information.
- **Eligible Profile Factors (interval only)**: These define the profile factors that are considered to be eligible for interval measuring components of this type. One profile factor can be identified as the default. The default profile factor will be automatically selected in system processing when a profile factor is required.
- **Valid Profile Factors for Conversion from Scalar to Interval (scalar only)**: These define the profile factors that are considered to be eligible for scalar measuring components of this type when converting scalar measurements to interval measurements. These profile factors are used to produce a curve of interval data from a scalar value. Without one of these factors defined scalar to interval conversion will use a flat line method (i.e. evenly divide the scalar value across the intervals). One profile factor can be identified as the default. The default profile factor will be automatically selected in system processing when a conversion profile factor is required.
- **Valid Scratchpad Measuring Component Types**: These define the scratchpad measuring component types considered valid for measuring components of this type.
- **Related Statistics Measuring Component Types**: These define the measuring component types that will be used to store statistical information about the historical usage of measuring components of this type. Please refer to Configuring Measuring Component Statistics for more information on how this list is used.
- **Display Properties**: Defines how measurement data for measuring components of this type is displayed, including:
  - *Display Configuration*: Details related to how measurements are displayed, including the 360 chart rendering method, number of hours of data to display, the maximum days to search for measurements, the default TOU map used, the TOU by Day Profile factor used, and default measurement condition.
  - *Event Bar Profiles*: The event bar profiles used when displaying measurement data for measuring components of this type. Event bar profiles are defined as values for the 360 View Event Bar Profile extendable lookup.
  - *Final Values Overlay Profiles*: The final values overlay profiles used when displaying measurement data for measuring components of this type. Final values overlay profiles are defined as values for the Final Values Overlay Profile extendable lookup.
  - *Measurement Conditions Not Shown on Chart*: The measurement conditions that should be omitted from rendering onto 360 Degree charts. Measurements whose conditions match these values will be rendered as gaps. For example, many 360 Degree charts use the condition "No Read - System" to represent the lack of a measurement, by adding this condition to this list it a gap will be rendered instead of a line with a 0 quantity measurement and a condition of "No Read - System".

When creating a measuring component type the following options are available:

| Name | Details | Business Object |
|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Interval Channel Type - Physical | Provides the configuration for a physical interval channel (including interval size). This is recommended for measuring components that measure consumptive usage in interval sizes up to 1 hour. | D1-IntervalChannelTypePhysical |
| Interval Channel Type - Scratchpad | Provides the configuration for a scratchpad interval measuring component. | D1-IntervalChannelTypeScratchp |
| Interval Channel Type - Physical Subtractive | Provides the configuration for a physical subtractive interval channel. In addition to standard interval configuration (including interval size) it also provides additional subtractive specific configuration (e.g. rollover validation, estimate reevaluation, etc). This is recommended for measuring components that measure subtractive usage in interval sizes up to 1 hour. | D1-IntrvlChanTypPhysSubtractiv |
| Interval Channel Type - External | Provide the configuration for interval measuring components that measure data not typically stored in the system, such as Amps. Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring components that don't exist. Filtering of data for external measuring components is configured via the **IMD Handling** flag. | D1-IntervalChannelTypeExternal |
| Register Type | Provides the configuration for a physical register that is manually read (e.g. rollover validation). These can be either consumptive or subtractive but are expected to be read infrequently (e.g. once a month). | D1-RegisterTypePhysical |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Auto-Read Register Type | Provides the configuration for a physical register that is automatically read. In addition to the standard register configuration (e.g. rollover validation) it also provides details around the schedule of expected readings (e.g. first daily measurement time and expected hours between measurements). These can either be consumptive or subtractive but are expected to be read frequently (e.g. at least once per day).<br>Note: these measuring component types can also be used to model interval data that is received with a large interval size (24 hours and larger). | D1-AutoReadRegisterType |
| Scalar Channel Type - External | Provide the configuration for scalar measuring components that measure data not typically stored in the system. Incoming data for external measuring components of this type can be filtered or passed through to other systems, preventing errors generated by incoming data for measuring components that don't exist. Filtering of data for external measuring components is configured via the **IMD Handling** flag. | D1-ScalarChannelTypeExternal |
| Aggregator Type | Provides the configuration for an aggregation measurement. | D2-AggregatorType |

> **Note**
>
> For “interval” data with an interval size greater than one hour, use the D1-AutoRegisterReadType measuring component type business object and the D1-RegisterAutoRead measuring component business object. This business object is well suited to daily and sub-daily readings. Using an interval data business object can result in measurement date problems during daylights saving time.

Refer to Configuring an Out-of-the-box Aggregation for more information about aggregation configuration

### Measuring Component Type Value Identifiers

Measuring components are configured to measure specific types of quantities this is defined by the list of value identifiers on the measuring component type. Up to eleven value identifiers can be specified. The primary measured quantity should be identified using the Value Identifier Type of Measurement. An additional ten derived values can be computed based on the primary

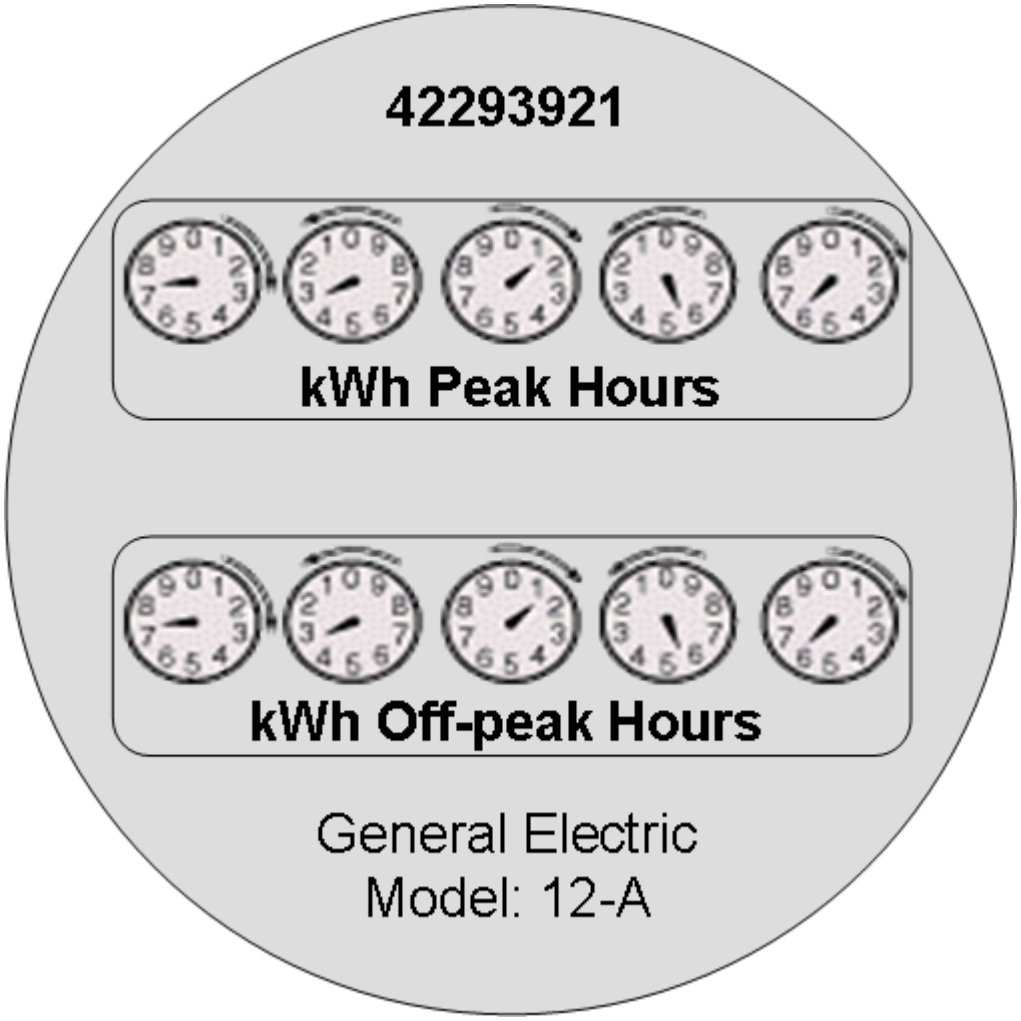

Copyright © 2026, Oracle and/or its affiliates.

measurement, these are identified with the Value Identifier Type of Value 01 through Value 10. Each value identifier is constructed of:

- **Unit of Measure**: The unit of measure for the quantity being recorded. Examples include kilo-watt hours (kWh), kilo-watts (kW), therms, cubic feet (CCF), temperature (Fahrenheit or Celsius), etc. Refer to Understanding Units of Measure for more information.
- **Time of Use**: Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc. Refer to Understanding Time of Use for more information.
- **Service Quantity Identifiers**: Used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Generally, SQI is only used when multiple measuring components measure the same thing, but in different ways. A meter that measures both generation KWH and consumption KWH could use SQIs to differentiate between the two. Refer to Understanding Service Quantity Identifiers for more information.
- **Value Derivation Algorithm**: Unlike UOM, TOU, and SQI this is not used in the identification of what is measured but rather is used to calculate a derived value based on the primary measurement. An algorithm from the list should be selected that contains the appropriate logic for calculating the derived value. This is applicable for those value identifiers with a Value Identifier Type of Value 01 through Value 10. More information on how derived values are calculated can be found in the Important Measuring Component Type System Events topic in this chapter. For more functional information about derived values please refer to About Final Measurements.

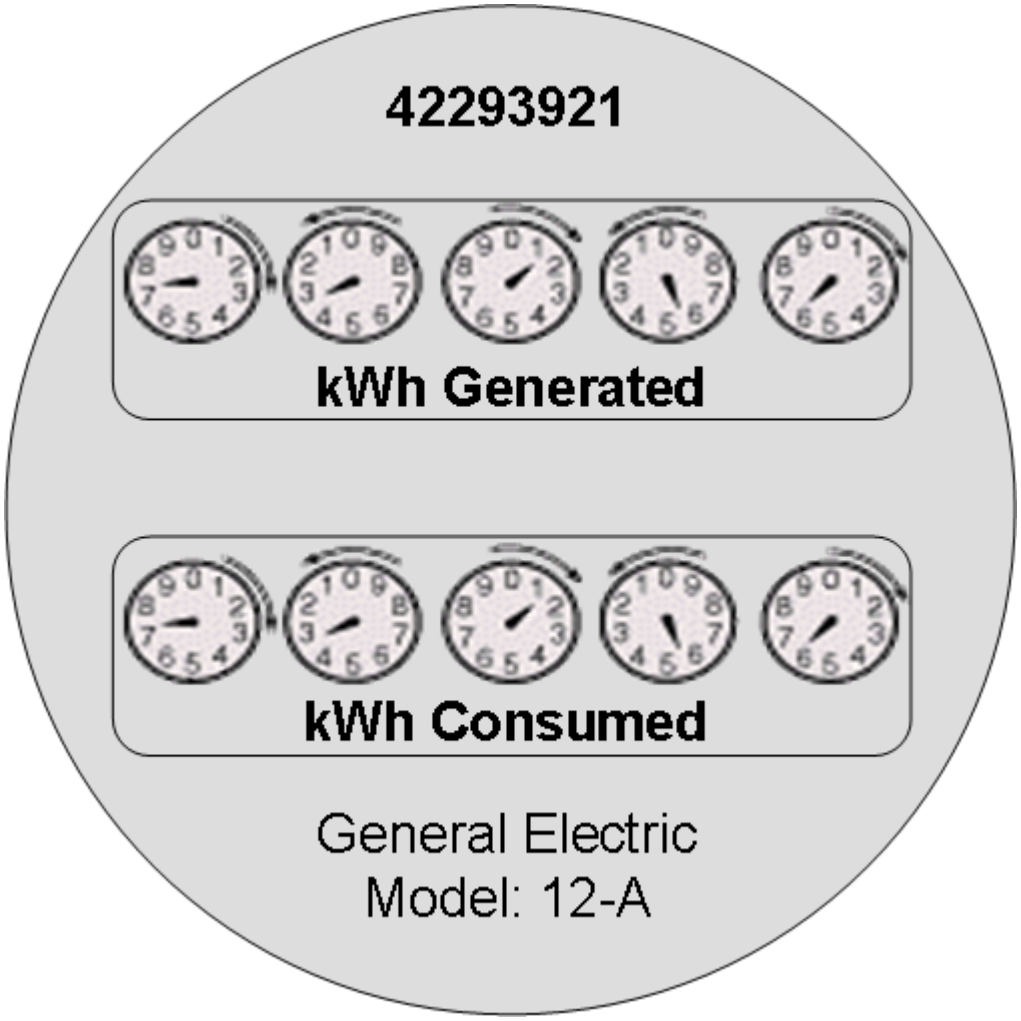
The combination of UOM, TOU and SQI define what a measuring component measures. TOU and SQI are optional, but UOM must be defined for all value identifiers. For example, consider a meter (as illustrated in the image below) with two measuring components, both measuring the same unit of measure (kWh), but each measuring component measures consumption in different time of use (TOU) periods (peak and off-peak).


Copyright © 2026, Oracle and/or its affiliates.




Another example might be a meter that records both generated KWH and consumed KWH. This meter would be configured to measure both UOM and SQI.


Copyright © 2026, Oracle and/or its affiliates.




A measurement is recorded each time a measuring component is read. This means that for a meter with two measuring components that is read once a month, two measurements, one for each measuring component, would be recorded each month.

**Important Measuring Component Type System Events**

The measuring component type supports several business object algorithm system events that relate to calculating the consumption for measuring components of that type:

- **Calculate Interval Consumption**: receives the interval list and performs any necessary calculations on that interval data to compute consumption. Since interval data is already received as consumption data algorithms for this system event are typically limited to application of the appropriate multipliers. Refer to the algorithm type Calculate Interval Consumption (D1-IN-CNSUMP) as an example.
- **Calculate Scalar Consumption**: receives information about the scalar reading and calculates the consumption as appropriate. Algorithms for this event typically support calculating the consumption using a stop and start reading or backing into a reading using consumption and a start reading. Much like the interval counterpart it will apply the appropriate multipliers. Refer to algorithm type Calculate Scalar Consumption (D1-SC-CNSUMP) as an example.
- **Calculate Subtractive Interval Consumption**: receives the interval list and supporting information (e.g. the start reading for the first interval) and either calculates the consumption by subtracting the interval's reading from the prior interval's reading or


Copyright © 2026, Oracle and/or its affiliates.

calculates the reading by adding the current interval's consumption to the prior interval's reading. Refer to algorithm type Calculate Subtractive Interval Consumption (D1-SIN-CNSUM) as an example.

- **Condition Mapping**: receives the details of a single subtractive interval along with the details for its start reading and computes the applicable final condition and reading condition. This is leveraged solely for subtractive interval measuring component types. Refer to algorithm type Subtractive Interval - Condition Mapping (D1-SIN-CNMAP) as an example.

These system events are typically called from within initial measurement processing during the initial stages of the initial measurement lifecycle (e.g. the Pre VEE status of most initial measurement business objects).

In addition to the measuring component type business object algorithms there is an additional system event provided on measuring component type itself:

- **Value Derivation**: receives details of an initial measurement and an associated final measurement. Using these inputs it can compute a value derived either from the primary measurement or one of the other derived values. Refer to algorithm type Derive a quantity using a formula (D1-DERIVAQTY) as an example.

#### Important Measuring Component System Events

The measuring component business object that is associated to a given measuring component type supports a special system event that is used in the periodic estimation process:

- **Periodic Estimation**: this system event will scan the measuring component's final measurement history to identify missing measurements and create either a To Do, or an estimation initial measurement, or both. More detail about this system event can be found by visiting the following algorithm types: Refer to algorithm type Create Interval IMD and To Do Based Upon Install History (D1-CIITBIH) and Auto-Read Scalar Periodic Estimation (D1-ARSPE) as an example.

#### Measuring Component Business Object Options

The device business object that is associated to a given device type plays an important role in how a device is processed in a system beyond defining the data associated to that device. Below are a list of business object options that are defined on the device business object and their impact on system processing:

- **Estimation Initial Measurement Data BO**: Is used to identify the appropriate estimation IMD to create.
- **Manual Override IMD BO**: Is used to identify the appropriate manual override IMD to create.
- **System IMD BO**: Is used to identify the appropriate system IMD to create.
- **Measuring Component Consumption Function**: identifies a function that can be executed from any compatible measuring component 360 Degree zone. Each measuring component can support 0 to many of these functions.
- **Interval Initial Measurement Function**: identifies a function that can be executed from any compatible initial measurement zone. Each measuring component can support 0 to many of these functions.

More detail about these options can be found by visiting a measuring component business object and inspecting the business object options.


Copyright © 2026, Oracle and/or its affiliates.

## Configuring Measuring Component Types

This portal is used to display and maintain a Measuring Component.

Refer to Understanding Measuring Component Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Measuring Component**.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Type:** This zone provides information about the selected Measuring Component Type.

# Measuring Component Comparison Types

- Understanding Measuring Component Comparison Types
- Configuring Measuring Component Comparison Types

## Understanding Measuring Component Comparison Types

Measuring component comparison types define the details by which measuring component data can be compared to determine the days that most closely resemble a specific day being evaluated. For example, measuring component comparison types can be used on measuring components that store weather data so that comparisons can be made based on temperature. These are used when performing “proxy day” estimations, allowing the application to search for days that are similar to historical data (for instance, the same date from the previous month or previous year ) to use as the basis for estimation.

Measuring component comparison types use the following parameters:

- A specific metric to be calculated to enable data comparisons. The “Calculate Comparison Coefficient - Pearson Correlation Coefficient” algorithm, included with the base package can be used to calculate a coefficient based on the linear correlation between two data sets (specifically between weather data stored on a measuring component and on the comparison object).
- The specific object to be compared. This can be either the same measuring component (to compare against historical data for the same measuring component), or a factor profile (to compare against profile data).
- The date range for the comparison period, including the start of the comparison period (current date, same date last year, or both), the time each day through which the comparison will be calculated, and the number of days prior to and after the start date that will be compared with the comparison object.
- Average Difference Eligibility: An average value either above or below the average for the current date that will be eligible for comparison. For example, assume the current date’s average is 75, and the prior day’s average is 72. If the Average Difference Eligibility is set to 3 then the prior day value of 72 will be eligible for comparison to the current day average of 75.
- The maximum number of resulting calculations that should be stored as measuring component comparable periods.


Copyright © 2026, Oracle and/or its affiliates.

- A template that defines which days are considered like each day of the week (for example, weekdays are like other weekdays, weekend days are like other weekend days). This template is defined as a record for the “Like Day Eligibility Template” extendable lookup.
- Work calendars used to determine if specific data should be compared to other weekend days or to the same date in prior years, and which dates, if any, should be excluded from historical data used for comparison.

## Configuring Measuring Component Comparison Types

Refer to Understanding Measuring Component Comparison Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Measuring Component Comparison Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Comparison Type List**: This zone lists all measuring component comparison type records. Broadcast a record to display the details of the selected record.
- **Measuring Component Comparison Type**: This zone displays details for the selected measuring component comparison type.

# Device Configuration Types

- Understanding Device Configuration Types
- Configuring Device Configuration Types

## Understanding Device Configuration Types

Device Configuration types define the measurements a given device can record. It should be noted that the device configuration type itself does not specify this information but rather the device configuration type will identify the valid list of measuring component types that can be referenced associated to a device.

When creating a new device configuration type there are the following options:

| Name | Details | Business Object |
|---|---|---|
| Device Configuration Type | This represents configurations for physical meters and communication components. It identifies a list of valid measuring components types. | D1-DeviceConfigurationType |
| Item Configuration Type | This represents badged items that are installed at a particular service point. Given the nature of items there are no measuring component types associated to this type. | D1-ItemConfigurationType |

Refer to Configuring Consumption Synchronization for more information on how a device configuration type can be set up to synchronize consumption between two related channels of measurement data.


Copyright © 2026, Oracle and/or its affiliates.

## Configuring Device Configuration Types

This portal is used to display and maintain a Device Configuration Type.

Refer to Understanding Device Configuration Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Device Configuration Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Configuration Type List:** This zone lists all Device Configuration Type records. Broadcast a record to display the details of the selected record.
- **Device Configuration Type:** This zone provides information about the selected Device Configuration Type.

# Device Types

- Understanding Device Types
- Configuring Device Types

## Understanding Device Types

Device types define information about a class of devices, including properties that apply to all devices of a type. Properties defined for a device type can be overridden for an individual device

Specifically, device types provide information about:

- How a device can be configured by defining the valid list of device configuration types
- How a device communicates by defining a fallback head end system and a list of valid head end systems.
- How a device records measurement data through the fallback incoming data shift which plays an important role in daylight savings processing. Refer to Daylight Savings Time Support for more information.
- The consumption profile of an item: the unit of measure consumed and method of calculation (profile or straight line)
- Whether the device is attached to a service point or meter (used with items and device communication equipment)
- Characteristics common to all devices of this type (defined in the **Device Type Characteristics** zone).

When creating a new device type there are the following options:

| Name | Details | Business Object |
|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Smart Meter Device Type | These devices measure consumption for a given service point, support remote commands, and remote collection of measurement and event data. These devices are associated to a head end system. | D1-SmartMeterType |
| Manual Meter Device Type | These devices measure consumption for a given service point but require a visit to the service point to collect the measurement data. They are not associated to head end systems. | D1-ManualMeterType |
| Item Device Type | These devices represent various items that consume consumption at a constant rate which are not directly metered. As consumption is calculated rather than measured so there is no need for an association to a head end system. Devices of this type can either be created individually for "badged" items or can be listed directly on a service point for "unbadged items."<br>These devices may also be configured as equipment that can be attached to a service point. | D1-ItemType |
| Communications Component Device Type | These devices provide remote collection of consumption data measured by a manual meter on which they are installed. These devices are associated to a head end system. | D1-CommunicationCompMeterType |
| Device Communication Equipment Type | These devices provide remote collection of consumption data measured by a manual meter to which they are attached. Devices of this sort are sometimes referred to as Encoding Receiver/Transmitter (ERT) modules, or communication modules.<br>These devices are associated to a head end system. | D1-DvcCommEquipmentType |

### Device Business Object Options Drive Functionality

The device business object that is associated to a given device type plays an important role in how a device is processed in a system beyond defining the data associated to that device. Below is a list of business object options that are defined on the device business object and their impact on system processing:


Copyright © 2026, Oracle and/or its affiliates.

- **Device Category**: This defines the type of device and correlates to the core device types that are supported (i.e. smart meter, AMR meter, manual meter, item, or communication component). This value is used by smart meter commands, service order management, and other system processes to make processing decisions.
- **Install Event BO**: This identifies the appropriate install event business object to use when this device is being installed at a service point. Refer to About Install Events for more information about install events.
- **Valid Command Request BO**: This identifies the commands that are valid for any device of this type. This option can be repeated for as many commands as the device supports. Note: the combination of the device and service provider (aka head end) define the true list of available commands by device.
- More detail about these options can be found by visiting a device business object and inspecting the business object options.

#### Device Type - Service Quantity

For "badged" and "unbadged" items consumption is not directly measured. Instead, for each item type, the average daily consumption amount is provided. The commodity represented by the daily consumption amount is defined by the item type's unit of measure.

The average daily consumption amount can be effective dated over time to support changes in the consumption profile. This is used in conjunction with the item's calculation method to derive the consumed amount per interval.

Refer to Multiple Time Zone Support for additional information on how device type service quantities are impacted by multiple time zones.

#### Valid Device Characteristic Types

The **Valid Device Characteristic Types** zone lists the valid characteristic types that may be instantiated on devices of this type. Characteristics of these types are maintained using the **Device Characteristics** zone in the **Device** portal.

## Configuring Device Types

This portal is used to display and maintain a Device Type.

Refer to Understanding Device Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **Device Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Type List:** This zone lists all Device Type records. Broadcast a record to display the details of the selected record.
- **Device Type:** This zone provides information about the selected Device Type.
- **Device Type Service Quantity:** This zone displays the list of effective dated service quantities supported by the selected Device Type. This applies only to Items.
- **Device Type Characteristics**: This zone lists characteristics common to all devices of this type.
- **Valid Device Characteristic Types**: This zone lists the valid characteristic types that may be instantiated on devices of the device type. To add or edit valid characteristic types, click **Edit**.


Copyright © 2026, Oracle and/or its affiliates.

# Defining Device Installation Options


- Service Point Types
- Service Point Quantity Types
- Measurement Cycles
- Measurement Cycle Schedules


## Service Point Types


- Understanding Service Point Types
- Configuring Service Point Types


### Understanding Service Point Types

Service point types define a specific type of point at which service is delivered.

Specifically, service point types define how the application manages many aspects of the service point's behavior. A service point type may have one or more valid device types defined that limit the types of devices that can be installed at service points of this type.

Service point types are defined by the following:

- **Service Point Business Object**: the business object used by service points of this type
- **Service Point Category**: the types of devices that can be installed at service points of this type. Valid values include:
  - **Meter**: Indicates that a single meter can be installed at service points of this type.
  - **Item**: Indicates that a single "badged" item can be installed at service points of this type.
  - **Multi-Item**: Indicates that one or more "unbadged" items can be installed at service points of this type.
- **Divisions**: one or more divisions associated with service points of this type.
- **Valid Device Types**: the device types considered valid for service points of this type.
- **Valid Field Task Types**: the field task types considered valid for service points of this type. The field task types available when creating field activities can be based on the field task types listed here via the **Restrict Field Task Type by Service Point Type** option on the Service Order Management Master Configuration.
- **Usage Snapshot Configuration**: the configuration(s) to be used to take weekly or monthly usage snapshot(s)
- **Unreported Usage Snapshot Configuration**: the configuration(s) to be used to take weekly or monthly unreported usage snapshot(s)

### Configuring Service Point Types

You use the Service Point Type portal to view and maintain service point types.



Copyright © 2026, Oracle and/or its affiliates.

Refer to Understanding Service Point Types for more information.

You can access the portal by selecting **Admin**, **Device Installation**, **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured service point types as well service point types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point Type List**: This zone lists all service point type records. Broadcast a record to display the details of the selected record.
- **Service Point Type**: This zone provides information about the selected service point type.
- **Valid Field Task Types**: This zone lists the field task types considered valid for the selected service point type.
- **Valid Disconnect Location**: This zone lists disconnection locations considered valid for the selected service point type.

# Service Point Quantity Types

- Understanding Service Point Quantity Types
- Configuring Service Point Quantity Types

## Understanding Service Point Quantity Types

Service point quantity types define types of quantities that can be stored for a service point. These quantity types are often infrequently calculated descriptors of a given customer and how they relate, at an aggregate level, to either the customer base as a whole or their particular rate class. For example, a customer quantity may be used to store a scaling factor that describes how a given customer’s usage compares to the profiled usage for their rate class.

In a settlement implementation, service point quantity types can be used to define values calculated monthly, annually, etc., such as annual peak load contribution (PLC).

Service quantity types use the following parameters:

- The service type (electric, gas, water, etc.) for service point quantities of this type
- Quantity identifiers related to the current service point quantity type (used to provide shorthand descriptions of the various types of values measured by service point quantities of this type)

## Configuring Service Point Quantity Types

Refer to Understanding Service Point Quantity Types for more information.

You can access the portal by selecting **Admin**, then **Device Installation**, then **Service Point Quantity Type**.


Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Service Point Quantity Type List**: This zone lists all service point quantity type records. Broadcast a record to display the details of the selected record.
- **Service Point Quantity Type**: This zone displays details for the selected service point quantity type.

# Measurement Cycles


- Understanding Measurement Cycles
- Configuring Measurement Cycles
- Measurement Cycle Processing


## Understanding Measurement Cycles

Measurement cycles define the schedule for manual meter reading of devices at Service Points in that cycle. Measurement cycles can have one or more associated routes used to collect measurements.

Measurement cycles can also be configured to define when to create usage transactions for Usage Subscriptions associated to Service Points in the cycle.

Refer to Measurement Cycle Processing for details on features, configuration and implementation.

## Configuring Measurement Cycles

This portal is used to display and maintain a Measurement Cycle.

Refer to Understanding Measurement Cycles for more information.

You can access the portal by selecting **Admin**, then **Device Installation**, then **Measurement Cycle**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measurement Cycle List**: This zone lists all Measurement Cycle records. Broadcast a record to display the details of the selected record.
- **Measurement Cycle**: This zone provides information about the selected Measurement Cycle.
- **Measurement Cycle Route List**: This zone lists the measurement cycle routes related to the measurement cycle. Click the **Route** link to view details for a measurement cycle route in the **Measurement Cycle Route** portal.
- **Measurement Cycle Schedule List**: This zone lists the measurement cycle schedules related to the current measurement cycle. Click the **Scheduled Selection Date** link to view details for a measurement cycle schedule in the **Measurement Cycle Schedule** portal.

### Configuring Measurement Cycle Routes

The Measurement Cycle Route portal is used to maintain individual measurement cycle routes.



Copyright © 2026, Oracle and/or its affiliates.

You can access the portal by clicking the **Route** for a specific measurement cycle route in the **Measurement Cycle Route List** zone of the **Measurement Cycle** portal. You are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measurement Cycle Route**: This zone provides information about the selected Measurement Cycle Route.
- **Measurement Cycle Route Characteristics**: This zone lists characteristics for the selected Measurement Cycle Route.

# Measurement Cycle Processing

Measurement cycle processing supports the following processes:

- **Meter Read Download** (for service point meters scheduled to be read for the measurement cycle): This process stages the activity for the extraction and download of the service point and meter information scheduled for reading.
- **Bill Determinant Push** (for subscribing external applications): This process stages usage transactions that calculate usage that will be sent to the subscribing external application. In this case, measurement cycles can be configured to define when to create usage transactions for Usage Subscriptions associated to Service Points in the cycle. Even Service Points whose meters are read automatically may reference measurement cycles. Creating bill determinants (by creating a usage transaction) is performed by an algorithm on the "Complete" state of the SP/Measurement Cycle Schedule Route business object (similar to creating activities as described above).

Measurement cycle processing is managed by the following three batch processes:

- **Create Pending Measurement Cycle Schedule Routes** (D1-CMCS): This batch process creates Schedule Routes for Measurement Cycle Schedules whose schedule selection date is on or before the batch business date. This process is used if routes have the same schedule each month, quarter, and so on. This process simply copies the routes from the Measurement Cycle to the Measurement Cycle Schedule on/after the scheduled selection date.
- **Create Pending SP / Measurement Cycle Schedule Route Records** (D1-CSPSR): This batch process creates a "SP/Measurement Cycle Schedule Route" transaction for every Service Point in the Measurement Cycle Schedule Route that is ready for processing.
- **Process Pending SP / Measurement Cycle Schedule Route Records** (D1-PSPSR): This batch process transitions the Pending "SP/Measurement Cycle Schedule Route" transactions to their Complete state. Algorithms can be configured to create "Meter Read Download" activities or usage transactions. These algorithms would be configured as an Enter algorithm on the "Complete" state of the SP/ Measurement Cycle Schedule Route business object (see below).

Measurement processing requires the following configurations:

- **Meter Read Download**: Configure the Create Meter Read Download Activity (D1-CRMRDACT) algorithm on the "Complete" state of the SP / Measurement Cycle Schedule Route (D1-SPMsrmtCycScheduleRoute) business object.
- **Bill Determinant Push**: Configure the Create Usage Transaction (D2-CRUSGTRN) algorithm on the "Complete" state of the SP / Measurement Cycle Schedule Route (D1-SPMsrmtCycScheduleRoute) business object.

Refer to About Usage Transactions for more information about usage transactions.


Copyright © 2026, Oracle and/or its affiliates.

# Measurement Cycle Schedules

- Understanding Measurement Cycle Schedules
- Configuring Measurement Cycle Schedules

## Understanding Measurement Cycle Schedules

Measurement cycle schedules define the dates on which devices are scheduled to be read for a given measurement cycle and the routes used to collect measurements for the measurement cycle.

For a deeper functional understanding, refer to the About Route Management section of the *Oracle Utilities Meter Solution Business User Guide*.

## Configuring Measurement Cycle Schedules

This portal is used to display and maintain a Measurement Cycle Schedule.

Refer to Understanding Measurement Cycle Schedules and Understanding Measurement Cycles for more information.

You can access the portal by selecting **Admin**, then **Device Installation**, then **Measurement Cycle Schedule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Measurement Cycle Schedule Query:** This zone allows you to query for Measurement Cycle Schedules based on Measurement Cycle and select the desired record.
- **Measurement Cycle Schedule:** This zone provides information about the selected Measurement Cycle Schedule.

# Defining Measurement Options

- Initial Measurement Data
- Understanding Interval Price Set Types
- Defining Interval Price Set Types
- Understanding IMD Types
- Configuring IMD Types

# Initial Measurement Data

- Configuring the Initial Measurement Algorithms
- Configuring Measurement Logging


Copyright © 2026, Oracle and/or its affiliates.

- Configuring Direct Measurement Processing
- Configuring Machine Learning Anomaly Scoring
- Removing Unwanted Initial Measurements

# Configuring the Initial Measurement Algorithms

The behavior of initial measurement data processing can be adjusted by updating parameters in several key algorithms in the initial measurement lifecycle. This includes algorithms on initial measurement business objects used with Legacy Measurement Processing and algorithms on IMD types used by Direct Measurement Processing.

> **Note**
>
> This is not a list of all algorithms and algorithm parameters that can be customized in the initial measurement processing lifecycle. Rather, it is a selection of some of the more impactful parameters.

**Initial Measurement Data Seeder Algorithms**

- Derive Service Provider and Measuring Component (D1-DER-SPRMC) / Derive Service Provider and Measuring Component Service (D1-DERSPRMCS):
  - Enabling a standalone measuring component search allows initial measurement data to be processed without a device identifier. This is useful when profile or temperature data is loaded through an IMD Seeder or seeder IMD type and a match must be made on measuring component identifiers alone.
  - If measuring component identifiers can be repeated across measuring components for the same device then the error for duplicate measuring components being found can be turned off. This is useful for certain head end systems that will have multiple channels with the same channel identifier. When a duplicate is found the search will fall back on other means for identifier the correct measuring component.
- Perform Date/Time Adjustments and Undercount/Overcount Check (D1-DODTTMADJ) / Perform Date/Time Adjustments and Undercount/Overcount Check Service (D1-DODTTMADS):
  - Undercount validation can be turned off completely or left enabled. When left enabled the automatic filling of gaps can be turned off separately.
  - The overcount check can be turned off.
  - The automatic adjustment of the individual interval date/times to an interval boundary can be turned on.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If your implementation receives initial measurement data with date/times that do not include an explicit time zone and the devices report date/times in standard time, you may need to add the following configuration:
>
> - Navigate to **Admin**, then **Integration**, then **Message Option**.
> - In the **XAI compliance** option, ensure that the following text is provided: `xsd:strict:dstGapInStandardTime`
>
> This will prevent a date/time reported in standard time that falls on the missing hour of the day Daylight Savings Time is entered from being misinterpreted.

**Specific Initial Measurement Data Algorithms**

- Normalize measurements (overwrite identical existing Measurements) (D1-NORM-IMD) / Normalize Measurements Service (overwrite identical Measurements) (D1-AUTO-NORS):
  - When the initial measurement data includes measurements that match exactly with the existing final measurements they can either be overwritten or skipped.
  - If logging of changes to final measurements is desired then it must be indicated as such in this algorithm. It is suggested to keep this turned off for high volume initial measurement data such as initial load. See Configuring Measurement Logging for more details.

> **Note**
>
> Refer to Configuring Consumption Synchronization for additional parameters that can be adjusted on specific initial measurement data algorithms.

# Configuring Measurement Logging

There are two components to logging changes to final measurements:

- **Initial Measurement Finalization**: the "normalization" algorithm on a particular initial measurement data business object will determine how final measurements are updated. For performance reasons certain types of initial measurement data (e.g. initial load) are delivered with a final measurement update method that will skip any logging. This can be controlled by parameters on the algorithm that implements the algorithm type Normalize Measurements for Initial Measurement Data (D1-NORM-IMD) or Normalize Measurements for Initial Measurement Data Service (D1-NORM-IMDS) algorithm types. Specifically, with Legacy Measurement Processing the "Create Measurement Log on Update (Y/N)" parameter should be set to "N" as creating these logs has a performance impact. For Direct Measurement Processing, the "Create Measurement Audit (Y/N)") parameter should be set to "Y".
- **Measurement Business Object**: there must be an algorithm for the audit system event configured. The base package delivers the algorithm type Add Measurement Log Record (D1-AMSRMTLOG). It is this algorithm that actually records the log entries.


Copyright © 2026, Oracle and/or its affiliates.

### Logging User Updates to Manual Initial Measurements

By default, manual edits made by users of these zones are not logged on the Log tab. Logging of manual edits to manual initial measurements can be enabled by adding a logging algorithm on an appropriate lifecycle state of the manual initial measurement business objects. The Log User Transaction (D1-LOGUSRTRN) base package algorithm can be used for this. This Enter algorithm is designed to be defined on the Initial state of the manual initial measurement business objects, but it can also be defined on any (non-transitory) Interim or Final state as well.

To ensure logging of any or all manual edits made to manual initial measurements, this algorithm should be specified on any state in which users will make manual edits. This will most often be the Pending or VEE Ready states, but could also include the Error, Exception, or Finalized states.

> ⚠ **Caution**
>
> When defining this algorithm, the user should exercise caution and determine if previous algorithms in the sequence within the state contain any form of transitioning logic that may inadvertently cause this algorithm to be bypassed.

# Configuring Direct Measurement Processing

Direct Measurement Processing is a method of processing incoming initial measurement data designed to reduce the volume of initial measurement records in the database. Rather than create and update initial measurement data records, Direct Measurement Processing creates temporary staging records (in the IMD Control Staging (D1_IMD_CTRL_STAGE) table) and processes initial measurement data in memory and creates final measurements in the Measurement (D1_MSRMT) table. With Direct Measurement Processing, records in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table are created only if an error is encountered during processing (including VEE processing). See Direct Measurement Processing in the *Business User Guide* for more information about Direct Measurement Processing.

Configuring the system for Direct Measurement Processing involves the following:

### IMD Types

IMD Types are configuration records used to define how initial measurements are processed using Direct Measurement Processing. IMD Types are defined for specific measuring component types in the the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) for each head end system. See Understanding IMD Types for more information about IMD Types.

### Configure Extendable Lookups

Extendable lookups are used with Direct Measurement Processing for payload processing as well as with Measuring Component Filtering.

### Payload Processing

To configure Payload Processing for use with Direct Measurement Processing, enable the **Use IMD Type** option on your SGG Payload Processing Configuration extendable lookup values. See Creating SGG Payload Processing Extendable Lookup Values for more information.

### Measuring Component Filtering - Pass Through


Copyright © 2026, Oracle and/or its affiliates.

The Head End Status to Condition Map (D1-HeadEndStatusToConditionMap) extendable lookup is used to map head end systems to head end system status code mapping extendable lookups. This allows Direct Measurement Processing to identify which status code mapping extendable lookup to use based on the head end system associated with the incoming measurement (either specified in the measurement or derived from the measuring component).

### Configure Head End System Processing Methods

To configure head end systems for use with Direct Measurement Processing, update the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) to specify the IMD Types to use for your measuring component types for each head end system. See Creating Head End Systems for more information.

If your implementation uses "pass through" external measuring components, the appropriate (interval or scalar) pass through IMD type should be specified for each "pass through" external measuring component type.

### Batch Processing

Batch processing is used with Direct Measurement Processing for payload processing, Measuring Component Filtering, and Measuring Component Ordering.

#### Payload Processing

Direct Measurement Processing receives initial measurement payloads via Payload Processing batch processes based on the following templates:

- Payload Processing Monitor Template (D1-PLPRM)
- Payload Processing Monitor Template with Export (D1-PLPSO)

See Creating Payload Processing Batch Controls for more information.

#### Measuring Component Filtering - Pass Through

When using Measuring Component Filtering, the Extract Passthrough IMDs In Error (D1-EPIMD) batch process can be used to process "Waiting for Extract" records in the Initial Measurement Data (D1_INIT_MSRMT_DATA) table. This batch process uses the following parameters:

- **Maximum IMDs Per File:** When extracting data, a new file will be created when this limit is reached. Note: this limit is applied per thread.
- **Stream Out File Location:** The file location or Object Storage bucket where initial measurement are posted for distribution to external systems. This should be a file path or the Value of the File Location Extendable Lookup value for the appropriate bucket (usage). See Referencing URIs for more information.
- **Stream Out File Name Prefix**: The prefix for files created in the Stream Out File Location. The file format will be <Stream Out File Name Prefix><batch run>_<thread>_<thread count>_<timestamp>{<_file sequence>}.json.gz. If the Stream Out File Location is set then this parameter is required.

#### Measuring Component Ordering

When using Measuring Component Ordering, the Process IMD Control Stage Data (D1-IMDCS) batch process can be used to process "Ready to Process" records in the IMD Control Staging table. This batch process uses the following parameters:

- **Value to restrict to IMD Control Stage Status**: Designates the status of records to process. When processing Measuring Component Ordering initial measurements, this should be set to "D1RD" ("Ready to Process").


Copyright © 2026, Oracle and/or its affiliates.

- **Enable sequence handling**: Designates (true or false) if initial measurements should be retrieved and ordered for a measuring component in ascending order by date time and create date time. When processing Measuring Component Ordering initial measurements, this should be set to "true".

See Measuring Component Ordering in the *Business User Guide* for more information.

#### Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

#### Configure Info Exception IMD Header Business Object

If you want Info Exception IMD Header records to be included in Initial Measurement Data Header Export data, the Create Measurement Extract Request for Info Exceptions (D1CMERINFOEX) algorithm must be configured as an "Enter" algorithm on the Finalized state of the Informational Exception Header (D1-InfoExceptionHeader) business object.

## Configuring Machine Learning Anomaly Scoring

Machine Learning Anomaly Scoring is a method of processing incoming initial measurement data designed to use Artificial Intelligence and Machine Learning neural network models to recognize "normal" versus anomalous measurement values, and only perform Validation, Editing, and Estimation (VEE) processing for measurements deemed to be anomalous.

Configuring the system for Machine Learning Anomaly Scoring involves the following:

NOTE: Machine Learning Anomaly Scoring relies on machine learning models that must be deployed as part of your cloud service implementation. Please contact your Oracle Customer Success Manager if you wish to use Machine Learning Anomaly Scoring.

#### Measuring Components and Measuring Component Types

Specific measuring components and/or measuring component types are enabled for Machine Learning Anomaly Scoring using the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

In addition, the anomaly threshold for those measuring components and measuring component types can be defined using the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** fields (respectively).

#### MDM Master Configuration

The Machine Learning Configuring section of the MDM Master Configuration defines how results coming back from machine learning scoring are processed.

- **Exception Type**: specifies the VEE exception to be created when the machine learning anomaly score is higher than the **Anomaly Score Threshold** or when there was an error during the scoring process.
- **Anomaly Score Threshold**: the limit for the anomaly score of any measurement within an initial measurement data. If the anomaly score (the difference between the Pre-VEE measurement value and the machine learning value) for one or more initial measurement data measurements is equal to or higher than this threshold then the initial measurement data will be considered anomalous and will be processed through VEE rules. Any initial


Copyright © 2026, Oracle and/or its affiliates.

measurements where all anomaly scores are below this threshold skip VEE processing and move to finalizing the initial measurement data to the measurement table.
Note that this can be overridden by the **Anomaly Score Threshold** or **Anomaly Score Threshold (Fallback)** fields on specific measuring components or measuring component types (respectively).

### Configure Payload Processing Extendable Lookups

To configure Payload Processing for use with Machine Learning Anomaly Scoring, you need to create two types of Payload Processing Extendable Lookup values:

- **Initial Payload Processing**: These are standard SGG Payload Processing Configuration (D1-SGGPayloadProcessing) extendable lookup values. You need to create one of these for each head end system from which you intend to process data using Machine Learning Anomaly Scoring. See Creating SGG Payload Processing Extendable Lookup Values for more information.
- **Machine Learning Payload Processing**: These are values for the SGG ML Payload Processing Configuration (D1-SGGMLPayloadProcessing) extendable lookup. You need to create one of these for each head end system from which you intend to process data using Machine Learning Anomaly Scoring. Values for this extendable lookup include the following:
  - **Payload Handler Class Name**: The Java class name for the processing handler to be used. This should be "com.splwg.d1.domain.sgg.ml.processing.MLResultPayloadHandler".
  - **Head-End System**: The head end system for which the configuration will apply. Select from the drop-down list.
  - **Load All ML Results**: A checkbox that indicates if Post VEE data should be created in the IMD regardless of anomalous status. This should be avoided in production environments as it increases the storage requirement for each Initial Measurement Data record which can lead to slower transaction processing.
  - **Payload Processing Result Type**: The type of payload to which the configuration applies. Machine Learning Anomaly Scoring is only supported for **Initial Measurements** at this time.
  - **Use IMD Type**: A checkbox that indicates if payload processing should use Direct Measurement Processing and IMD Types. See Direct Measurement Processing in the *Business User Guide* for more information.
  - **Dynamic Parameters**: A list of parameter names and values that support additional processing rules associated with Machine Learning Anomaly Scoring. These include:
    - **boStateConditionCode**: Optional value. Specifies the business object State Condition (from the State Condition (F1_STATE_CONDITION_FLG) lookup) that the payload handler will check for in the initial measurement's current state before proceeding with processing of the payload. The base package includes the Waiting for ML (D1WM) State Condition defined for the “Waiting for ML” state on the Initial Load IMD business objects. Specifying this parameter with this value ensures that only initial measurements in the "Waiting for ML" state will be processed. If this parameter and value are set and the initial measurement is in some other state, processing stops and an error is written to the log.

### Configure Post-ML VEE Processing (Optional)

You can define VEE Groups that are executed after machine learning processing to compare or adjust the measurement values based on a related measuring component, such as a scalar measuring component defined as the **Consumption Reference Measuring Component** on


Copyright © 2026, Oracle and/or its affiliates.

an interval measuring component. These groups are referred to as "Post-ML" VEE groups, and are defined for the "Post-ML" VEE Role on measuring component types and measuring components.

Post-ML VEE Groups can include only the following VEE Rules:

- Sum Check
- Interval Adjustment From Scalar
- Scalar Calculation from Interval
- Final Measurement Replacement

To configure Post-ML VEE processing you need to do the following:

- Create one or more Post-ML VEE Groups containing one or more of the VEE rules supported for post-ML processing (listed above). Note that Post-ML VEE Groups can NOT reference Execute VEE Group or VEE Group Matrix rules.
- Add the Post-ML VEE Groups to the appropriate measuring component types and/or measuring components in the **Post-ML** VEE role. Note that Post-ML VEE Groups can only be added when machine learning is enabled for the measuring component or measuring component type via the **Enable ML Based Validation** or **Enable ML Based Validation (Fallback)** flag (respectively).

#### Batch Processing

Batch processing is used with Machine Learning Anomaly Scoring for both initial payload processing as well as processing data returned from the machine learning model. See Creating Payload Processing Batch Controls for more information about creating Payload Processing batch processes.

#### Initial Payload Processing

Machine Learning Anomaly Scoring receives initial measurement payloads via Payload Processing batch processes based on the following templates:

- Payload Processing Monitor Template with Export (D1-PLPSO)

This is used to send initial measurements to Object Storage for machine learning processing.

#### Processing Data Returned from the Machine Learning Model

Machine Learning Anomaly Scoring processing data returned from the machine learning model using Payload Processing batch processes based on the following templates:

- SGG ML Payload Processing Monitor Template (D1-PLMLP)

This is used to retrieve measurements following machine learning processing, populate the IMD with machine learning values and the Anomaly Score Threshold, and calculate the Anomaly Score for each measurement value. Batch processes based in this template require an SGG ML Payload Processing Extendable Lookup value.

- Historical Data Extracts
- Machine Learning Comparative Reporting

## Historical Data Extracts

When using Machine Learning Anomaly Scoring with interval measurements, the measurement data being processed includes enough historical data for the machine learning models to perform the anomaly scoring calculations, but when using Machine Learning


Copyright © 2026, Oracle and/or its affiliates.

Anomaly Scoring with scalar measurements, there is a chance that the models will not have access to sufficient historical data to perform those calculations.

In this case, you can extract historical data to supplement the cache of historical data used in the anomaly scoring calculations. This process supports extracts of both interval and scalar data.

### Required Configuration

Extracting historical data requires that the **Enable ML-Based Validation** flag be set yo "YES" for all measuring component type for which data is to extracted.

### Batch Controls

Extract of data and creation of extract files is performed by the Initial Extract for ML Cache (D1-INIML) batch process.

This batch process uses the following parameters:

- **Start Date/Time**: The start date and time of the data extract
- **Number of Days**: The number of days to extract (the process creates a separate file for each day)
- **Interval/Scalar**: Flag indicating which type of data to extract, interval (D1IN) or scalar (D1SC)
- **File Path**: The file path to where the process will post the file so that it can be ingested into the historical data cache used by the machine learning models. Refer to **Referencing URIs** (LINK) in the user documentation for more details.
- **File Limit**: The number of initial measurements to be included in the file (default is 10,000)

This process extracts data for ALL measuring components where the **Enable ML-Based Validation** flag on the Measuring Component Type is set yo "YES".

### File Creation and Naming

Because this process can potentially create very large extract files, the files are created with a controllable sizes. Data is extracted in the Adapter Development Kit Native Format, and each file will only contain initial measurements for a single day, and for a default of 10,000 initial measurements. The logic for creating files differs slightly between interval and scalar data:

- Scalar: Each measurement is an initial measurement. For the common scenario there will be one initial measurement per day. However, some customers get scalar data more often per day or some measuring components may have adhoc reading (field or on demand read command based). For those less common scenarios there may be many initial measurements for a single measuring component for a single day.
- Interval: Files contain and initial measurement for each calendar day. Since each initial measurement can contain a variable number of measurements, there will only ever be one initial measurement per measuring component per day.

Extract files are named based on the following syntax:

pre-vee-init-<channel type>-<date being extracted>-<thread number>-<file number>-<batch run>.xml.gz.

where:

- <channel type> is the type of data to extract: interval (D1IN) or scalar (D1SC)


Copyright © 2026, Oracle and/or its affiliates.

- <date being extracted> is the date for the data being extracted, in YYYYMMDDHH24MISS format YYYY = year, MM = month, DD = day, HH= hour, MI= minute, SS = second)
- <thread number> is the thread number that created the file
- <file number>: is the number of the file, in cases where multiple files are created from same batch process run
- <batch run> is the batch number for the process

## Machine Learning Comparative Reporting

Machine Learning (ML) Comparative Reporting enables you to compare ML anomaly results with Validation, Editing, and Estimation (VEE) results and other corrective actions. You can use this functionality to evaluate anomaly score thresholds before deployment and to monitor the effectiveness of ML anomaly scoring in production.

Before configuring comparative reporting, configure Machine Learning Anomaly Scoring, including the required machine learning model, payload processing, and batch controls. This includes set up and configuration of Oracle Cloud Object Storage Buckets (see Creating Object Storage Locations for more information).

Machine Learning (ML) Comparative Reporting can be used in two ways:

- **Historical Evaluation** evaluates existing interval measurement data (IMDs) processed using VEE for a selected period.
- **Ongoing Evaluation** captures comparison statistics for ML exceptions created during normal processing.
- Configuring and Running Historical Evaluations
- Configuring and Running Ongoing Evaluations
- Reviewing Comparative Results

### Configuring and Running Historical Evaluations

You can use historical evaluations to score existing initial measurement data (IMDs) originally processed using VEE using ML, compare the ML results with original VEE results, and determine an appropriate anomaly score threshold.

This process use batch processes to create and extract historical "verification" data that can be evaluated by ML and then compared to the original VEE data, Another batch process captures the results in a statistics control record, which can be viewed using the **Meter ML and VEE Dashboard** portal.

> **Note**
>
> Historical verification data is used only for coordinating the scoring by the ML model and evaluation. This data does NOT create or update any measurement data.

> **Note**
>
> Historical evaluation should be used in non-production environments (such as DEV or TEST) ONLY.


Copyright © 2026, Oracle and/or its affiliates.

### Configuring Historical Evaluations

Configuring historical evaluation involves the following:

#### Create ML VEE Exception Type

Define a VEE Exception Type that specifies the VEE exception to be created when the machine learning anomaly score is higher than the **Anomaly Score Threshold** or when there was an error during the scoring process when used with historical evaluation. This may be a different VEE exception Type than used for ongoing ML processing. See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

#### Update MDM Master Configuration

Update the **Machine Learning Configuration** section of the MDM Master Configuration as follows:

- **Exception Type**: Specify the ML VEE Exception Type created for ML historical evaluation (above).
- **Anomaly Score Threshold**: 0 (this ensures that the system retains the ML prediction and anomaly score for all IMDs selected for the evaluation)

See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

#### Configure Consumption Extract

Configure a Consumption Extraction Type for each ML model to be tested based on the ML Historical Verification Extract Type (D1-MLVerificationExtractType) business object. This construction extract type is used to define what is going to be scored as well as how VEE results should be interpreted as compared to ML results.

This should include the following attributes that define the model:

- **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
- **Measurement Frequency**: The frequency at which measurements are captured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
  - **For interval data**: This is the interval size.
  - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).
- **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring component types.
- **Service Point Types**: One or more service point types and UOM/TOU/SQI combination eligible for ML verification. These combinations determine which historical measuring components are evaluated.
- **VEE Exception Type Inclusions**: One or more VEE exception types that represent consumption quality issues ML is expected to detect. When this list is populated, any matching exception qualifies the IMD as having a VEE detected issue, regardless of severity. When empty, any non-informational VEE exception qualifies. You should exclude


Copyright © 2026, Oracle and/or its affiliates.

conditions that ML cannot evaluate so as to ensure ML to VEE comparisons remain meaningful.

- **UT Exception Type Inclusions**: One or more Usage Transaction exception types that indicate user intervention for an IMD with a qualifying VEE exception. The process compares the IMD with Usage Transactions for its Usage Subscription whose periods overlap the IMD period. A qualifying exception is interpreted as evidence of user action to resolve the IMD's data quality issue. When the list is populated, any matching exception counts regardless of severity. When empty, any non-informational Usage Transaction exception counts. Include only exception types that represent data quality issues so that comparisons between ML results and inferred user actions remain meaningful.

See About VEE and Understanding Exception Types for more information.

**Configure Statistics Control**

Configure a Statistic Control for each ML model to be tested based on the ML Historical Verification Control (D1-MLVerificationControl) business object. Snapshots created based on this statistics control compare the results from ML scoring with the previous VEE results, identifying when they agreed/disagreed. The process attempts to include user actions performed on VEE exceptions to determine if VEE exceptions were true issues or false positives.

These statistic controls are defined by the following:

- **Retention Mode**: Defines how to retain data for snapshots created from this statistic control
- **Snapshot Start Date**: The date on which snapshots created from this statistic control start.
  - **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
  - **Measurement Frequency**: The size of the intervals measured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
    - **For interval data**: This is the interval size.
    - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).
  - **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring-component types.
- **ML Threshold Override**: Use to override the **Anomaly Score Threshold** on the Master Configuration if applicable. This is not used with this historical evaluations.
- **Snapshot Business Object**: The business object to use when creating snapshots based on this statistic control. This should be ML Verification Statistics (D1-MLVerificationStatistics)

**Configure SGG ML Payload Processing**

Configure ML payload processing as described in Configure Payload Processing Extendable Lookups in Configuring Machine Learning Anomaly Scoring to process results returned from the ML model.

### Running Historical Evaluations

Running historical evaluations involves the following:


Copyright © 2026, Oracle and/or its affiliates.

### Create Historical Evaluation IMDs

Run the following batch processes in sequence to create "historical validation" initial measurement data that can be used for evaluation by running the following batch processes in sequence:

- Generate ML Historical Verification IMDs (`D1-GMLVI`) to select and create IMDs for evaluation. The IMDs are placed in the Waiting for ML state.
- Extract ML Historical Validation IMDs (`D1-EXHVI`) to export the selected IMDs for ML scoring.

Historical Verification IMDs:

- Are created only from "initial load" initial measurements,
- Duplicate the preVEE data which is sent to ML,
- Duplicate the VEE exceptions for statitsics computation
- Link to the original IMD for traceability

### ML Processing

The machine learning model scores the data and sends the results to an object storage location (similar to the process used with Machine Learning Anomaly Scoring

### Run ML Payload Processing and Analysis

Once the ML scoring results have been returned to object storage, run the following processes in sequence to process the payloads and capture information about the evaluation:

- SGG ML Payload Processing Monitor Template (`D1-PLMLP`) to process the returned ML results.
- IMD Monitor - Waiting for ML IMDs (`D1-IMDML`) to capture evaluation information and move the IMDs to the Analysis in Progress state.

### Capture Statistics

Open the Historical Statistics Control, and click **Capture**. When the dialog opens, enter the **Start Date** and the **Number of Days** to process and click **OK**. This creates a historical ML Verification Statistics Snapshot for each requested day using the statistics control's **ML Threshold Override** or the **Anomaly Score Threshold** defined on the MDM Master Configuration.

### Process ML Historical Statistics Snapshots

Run the ML Verification Statistic Snapshot Monitor (D1MLVSSM) batch process to calculate daily ML Verification Statistics Snapshots. The process selects ML Historical Validation IMDs based on service type, interval/scalar mode, and frequency, then stores ML, VEE, user comparisons and anomaly-score distributions for dashboard analysis.

### Re-evaluating Historical Results

You can make changes to the Consumption Extract Type or Statistics Control to fine tine the results and identify what works best for your implementation.

If you change the VEE Exception Type or Usage Transaction Exception Type configuration in the Consumption Extract Type:

- Run the Transition ML Historical Validation IMDs (`D1-TMHVI`) batch with the Transition Condition parameter set to `"D1RC"`


Copyright © 2026, Oracle and/or its affiliates.

- Run the ML Verification Statistic Snapshot Monitor (D1MLVSSM) batch process again and review the updated results.

If you only change the **ML Threshold Override** on the Statistic Control:

- Perform the **Capture** action again
- Run the ML Verification Statistic Snapshot Monitor (D1MLVSSM) batch process again and review the updated results.

This process can be run in stages:

- Stage1: ML scoring, run once
- Stage 2: Capture statistical inputs, which involves running `D1-TMHVI`), generating new snapshots, and running `D1MLVSSM` to calculate statistics
- Stage 3: Anomaly score evaluation, which involved creating snapshots and running statistics collection to see how the statistics change and to identify the right threshold

#### Removing Historical Evaluation Data

After completing a historical evaluation, run Transition ML Historical Validation IMDs (`D1-TMHVI`) batch processing with the Transition Condition parameter set to "`D1CN`" to complete the IMDs and remove their IMD control records.

To remove historical evaluation IMDs that were created in error, run the Remove ML Historical Validation IMDs (`D1-RMHVI`). batch process.

> **Note**
>
> Removing historical evaluation data is not mandatory, but if historical verification is being run on large amounts of data over a long period of time removing periods where research is complete help ensures that performance of the various processes does not degrade.

## Configuring and Running Ongoing Evaluations

You can use ongoing evaluation to collect comparative reporting statistics during normal ML anomaly scoring processing.

> **Note**
>
> It is not expected that historical and ongoing evaluations would be configured and active for the same time period. Ongoing is intended for tracking live production ML results, whereas historical evaluations are intended for data that has already been processed using VEE.

#### Configuring Ongoing Evaluation

Configuring ongoing evaluations involves the following:

**Create ML VEE Exception Type**

Define a VEE Exception Type using the ML Exception with Statistics business object (`D1-MLExceptionWithStats`). This specifies the VEE exception to be created when the machine learning anomaly score is higher than the **Anomaly Score Threshold** or when there was an


Copyright © 2026, Oracle and/or its affiliates.

error during the scoring process when used with ongoing evaluation. See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

> **Note**
>
> The ML Exception with Statistics business object (`D1-MLExceptionWithStats`) lifecycle delays capturing of statistics until after the production usage loading batch window so it can be done in off times during the day. The VEE exception itself is what is capturing the key inputs to the statistics calculations

**Update MDM Master Configuration**

Update the **Machine Learning Configuration** section of the MDM Master Configuration as follows:

- **Exception Type**: Specify the ML VEE Exception Type created for ML ongoing evaluation (above).
- **Anomaly Score Threshold**: The fallback threshold used when a threshold is not defined for the measuring component or measuring component type.

See MDM Master Configuration in Configuring Machine Learning Anomaly Scoring for more information.

**Configure Consumption Extract**

Configure a Consumption Extraction Type for each ML model to be tested based on the ML Historical Verification Extract Type (D1-MLVerificationExtractType) business object. Extracts created based on this type are used by the VEE Exception to capture the statistic inputs (such as which VEE exceptions to consider).

This should include the following attributes that define the model:

- **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
- **Measurement Frequency**: The frequency at which measurements are captured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
  - **For interval data**: This is the interval size.
  - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).
- **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring-component types.
- **Service Point Types**: One or more service point types and UOM/TOU/SQI combination eligible for ML verification. These combinations determine which historical measuring components are evaluated.
- **VEE Exception Type Inclusions**: One or more VEE exception types that represent consumption quality issues ML is expected to detect. When this list is populated, any matching exception qualifies the IMD as having a VEE detected issue, regardless of severity. When empty, any non-informational VEE exception qualifies. You should exclude


Copyright © 2026, Oracle and/or its affiliates.

conditions that ML cannot evaluate so as to ensure ML to VEE comparisons remain meaningful.

- **UT Exception Type Inclusions**: One or more Usage Transaction exception types that indicate user intervention for an IMD with a qualifying VEE exception. The process compares the IMD with Usage Transactions for its Usage Subscription whose periods overlap the IMD period. A qualifying exception is interpreted as evidence of user action to resolve the IMD's data quality issue. When the list is populated, any matching exception counts regardless of severity. When empty, any non-informational Usage Transaction exception counts. Include only exception types that represent data quality issues so that comparisons between ML results and inferred user actions remain meaningful.

See About VEE and Understanding Exception Types for more information.

**Configure Statistics Control**

Configure a Statistic Control for each ML model to be tested based on the ML Ongoing Verification Control (D1-MLOngoingVerificationCntrl) business object. This is similar to the statistics control created for historical evaluation, except that it uses the VEE exception as the source of the statistical inputs.

These statistic controls are defined by the following:

- **Snapshot Frequency Hours**: The frequency (in hours) at
- **Retention Mode**: Defines how to retain data for snapshots created from this statistic control
- **Interval Scalar**: A flag that indicates whether Interval or Scalar data is to be evaluated. Together with **Measurement Frequency** and **Service Type**, this helps define eligible measuring-component types.
- **Measurement Frequency**: The size of the intervals measured by the measuring component, represented as hours:minutes:seconds (00:00:00). Together with **Interval Scalar** and **Service Type**, this helps define eligible measuring-component types.
  - **For interval data**: This is the interval size.
  - **For scalar data**: This should align with the expected hours between measurements defined on the scalar measuring component type (such as 24:00:00 for daily, 08:00:00 for 3 times per day).\
- **Service Type**: The Service Type for the measurement data being evaluated (such as Electric, Gas, and so on). Together with **Interval Scalar** and **Measurement Frequency**, this helps define eligible measuring-component types.
- **Snapshot Business Object**: The business object to use when creating snapshots based on this statistic control. This should be ML Verification Statistics (D1-MLVerificationStatistics)

**Configure SGG ML Payload Processing**

Configure ML payload processing as described in Configure Payload Processing Extendable Lookups in Configuring Machine Learning Anomaly Scoring to process results returned from the ML model.

### Running Ongoing Evaluation

Running ongoing evaluations involves the following:

**Batch Processing**

Schedule the following batch processes to run on a regular basis:


Copyright © 2026, Oracle and/or its affiliates.

- 
- Payload Processing Monitor Template with Export (`D1-PLPSO`) to process incoming measurement data and sends it to ML for scoring.
- SGG ML Payload Processing Monitor Template (`D1-PLMLP`) to process the returned ML results
- Process IMD Control Stage Data (`D1-IMDCS`) to process records in the IMD Control table, including:
  - Running appropriate VEE processing based on the ML result for each IMD with measuring component ordering.
  - The results are either IMDs created in error, a header IMDs created for exceptions, or published measurements
  - This is when the VEE exception will be created for ML
- VEE Exception Monitor (`D2-VEEDF`) to transition all pending VEE exceptions and create the appropriate statistic characteristics against the IMD associated to the VEE exception
- Statistics Control Monitor (`F1-STATS`) to create snapshots that are available for review on the Meter ML and VEE Dashboard.

### Reviewing Comparative Results

You can use the **Meter ML and VEE Dashboard** to:

- Review anomaly-score distributions.
- Select an anomaly score threshold that provides a manageable number of exceptions for investigation.
- Compare ML exceptions with VEE results.
- Investigate cases where ML identifies an exception that VEE does not identify.
- Drill into individual IMDs to assess whether the selected threshold identifies meaningful issues.

You access this portal by selecting **Main Menu**, then **Dashboards**, and then **Meter ML and VEE Dashboard**. This portal includes the following zones:

- **ML Meter Statistics Snapshots Header**: used to define filters for data analysis
- **ML and VEE Comparison Matrix**: Compares the results of VEE rules against ML model predictions based on the current ML threshold.
- **ML and User Comparison Matrix**: Compares the actions users have taken on IMDs with VEE exceptions against ML model predictions based on the current ML threshold.
- **VEE and User Comparison Matrix**: Compares the actions users have taken on IMDs with VEE exceptions against VEE results.
- **User Actions**: Displays the conclusions drawn from the user's actions
- **Anomaly Score Distribution**: Displays the count of IMDs for a given anomaly score range

Embedded help icons provide additional details for each zone.

## Removing Unwanted Initial Measurements


Copyright © 2026, Oracle and/or its affiliates.

Over time, implementations can accumulate unnecessary non-final initial measurements which effectively become "stuck" in the system. Examples of these types of initial measurements include:

- Erroneously imported duplicate initial measurements
- Initial measurements that failed validation and were not corrected or replaced
- Initial measurements that triggered service issue monitors that remain unresolved

These initial measurements can also have related VEE Exceptions, To Do Entries, and Service Issue Monitors that remain "stuck" as well. This unnecessary data can impact overall system performance and should be reviewed and removed from time to time.

The Remove Unwanted Initial Measurements feature can be used to identify and close or delete these unwanted non-final initial measurements and any related VEE exceptions, To Do Entries and service issue monitors. This process is initiated by creating Request records which specify the parameters used to locate the non-final initial measurements to be removed. The request in turn initiates a batch process which identifies non-final measurements and related VEE Exceptions, To Do Entries, and/or Service Issue Monitors.

This functionality uses the following system objects:

| Object Type | Object Name | Description |
|---|---|---|
| Request Type Business Object | Remove IMDs Request Type (D1-RemoveIMDsType) | Used to define types of initial measurements removal requests.<br>The Remove IMDs Request (D1-RemoveIMDsRequest) request business object (see below) is defined as the "Related Transaction BO" for this business object, meaning that requests created for request types based on this business object use the Remove IMDs Request (D1-RemoveIMDsRequest) request business object. |
| Batch Control | Remove Non-Final IMDs (D1-RENFI) | Used to identify non-final measurements and related VEE exceptions, To Do Entries and/or Service Issue Monitors (Service Tasks). |
| Request Business Object | Remove IMDs Request (D1-RemoveIMDsRequest) | Used to create and process initial measurement removal requests. |
| Fact Business Object | Remove IMD Object Count (D1-RemoveIMDOjbectCount) | Used to capture details of the initial measurements and related objects to be removed. |

### Remove Unwanted Initial Measurements Request Types

Requests used to remove initial measurements are based on request types which specify the batch control used to identify and remove the initial measurements.

To create request types used with this feature:

1. Select Menu, **Admin**, **General**, **Request Type**, **Add**.


Copyright © 2026, Oracle and/or its affiliates.

2. Select "Remove IMDs Request Type" from the **Request Type Business Object** drop-down list, and click **OK**.
3. Enter the following:
    - **Request Type**: A code for the request type
    - **Description**: A description of the request type
    - **Batch Control**: Search for and select the batch control to be used for requests of this type. The default is Remove Non-Final IMDs (D1-RENFI), but implementations can create custom versions of this batch control.
4. Click **Save**.

#### Remove Unwanted Initial Measurements Requests

To create an initial measurement removal request:

1. Select **Menu**, **Tools**, **Request**, **Add**.
2. Select the appropriate request from the **Request Type** drop-down list and click **OK**.
3. Enter the **Submission Parameters** for the request. These include:
    - **Mode**: The mode in which the process will be run. Options include:
        - **Preview**: Returns a count of non-final initial measurements and related objects (VEE Exceptions, To Do Entries, and Service Issue Monitors)
        - **Close**: Transitions non-final initial measurements to the "Discarded" state and completes and/or closes related objects. Details regarding the data impacted by "Close" requests are captured and displayed in the **Object Log** zone.
        - **Delete**: Deletes non-final initial measurements any related objects. Details regarding the data impacted by "Delete" requests are captured and displayed in the **Object Log** zone.
    - **Thread Count**: The number of processing threads allocated to the process
    - **IMD Business Object**: Used to restrict the initial measurements processed to those of a specific business object
    - **IMD Status**:Used to restrict the initial measurements processed to those in a specific non-final state, including Pending, Held, Additional Mapping, Mapping Error, VEE Ready, Defer, Error, VEE Complete, Exception, or Force Complete
    - **IMD Create Date/Time From**: Required. The start date and time of the time period in which initial measurements to be removed were created. For example, to remove initial measurements created between January 1, 2023 and January 31, 2023, the **IMD Create Date/Time From** would be "01-01-2023 12:00 AM".
    - **IMD Create Date/Time To**: Required. The end date and time of the time period in which initial measurements to be removed were created. For example, to remove initial measurements created between January 1, 2023 and January 31, 2023, the **IMD Create Date/Time To** would be "01-31-2023 12:00 AM".
    - **Device Configuration**: Used to restrict the initial measurements processed to those related to a specific device configuration.
4. Click **Save**. The request will then open in the **Request** portal.
5. Click **Submit Batch** to process the request.

Initially requests are created in the "Pending" status. When the request is run (by clicking **Submit Batch**), a batch process based on the batch control defined on the request type is executed.


Copyright © 2026, Oracle and/or its affiliates.

- If the batch process errors, the request **Status** is set to the "Batch Error".
- If the batch process completes, the request **Status** is set to the "Completed".

Details of the process are captured in the **Submission Information** section, and the **Object Log** zone.

The **Submission Information** section includes:

- **Batch Submission**: Link to the batch job submission for the batch process. Click the link to navigate to the **Batch Job Submission** portal to view details of the batch process.
- **IMD SDE Extract Through Date/Time**: The date and time through which specialized data extracts have been performed for initial measurements impacted by the process.
- **End of Batch Date/Time**: The date and time when the batch process was completed.

The **Object Log** zone contains table with details regarding the initial measurements and related objects processed, including:

- **Object**: The type of object processed (Initial Measurement Data, Service Task, To Do Entry, or VEE Exception)
- **Specialized Data Extract Occurred**: An indicator as to whether or not ("Yes" or "No") occurred for initial measurement data
- **Count**: The number of objects processed.

Example:

| Object | Specialized Data Extract Occurred | Count |
|---|---|---|
| Initial Measurement Data | No | 434 |
| Initial Measurement Data | Yes | 134 |
| Service Task | | 1 |
| To Do Entry | | 425 |
| VEE Exception | | 413 |

### Batch Control

The Remove Non-Final IMDs (D1-RENFI) batch control identifies and closes or deletes unwanted initial measurements and their related objects. When run, this batch process creates a fact (using the Remove IMD Object Count (D1-RemoveIMDObjectCount) business object) that captures details regarding the batch process and the objects being removed during the process. These details are viewed in the **Object Log** zone in the **Request** portal.

Note that the Request ID is a required parameter for this batch control, meaning that this batch control cannot be run on its own. It can be only run from an initial measurement removal request.

Implementations can create specific versions of this batch control for use when removing specific types of data. For instance, an implementation could create a "Preview" version of this batch control, or a version used only with interval initial measurements.

Implementations can alter the processing of this batch control by creating alternate algorithms for each of the System Events used by this batch process. Refer to **Ad-hoc Processes** in the **Plug-in Driven Background Processes** section for more information.


Copyright © 2026, Oracle and/or its affiliates.

### Recommendations and Warnings

- We recommend using the **Preview** mode to identify potential initial measurements for removal before processing a request in either the **Close** or **Delete** mode. This allows you to review and verify the data to be removed before closing or deleting it.
- We recommend that access to this feature be limited to system administrators and other power users. Access to this feature can be restricted via the Remove IMDs Request BOAS (D1-RMIMDREQBOAS) application service.

> **Warning**
>
> If used improperly, this feature can inadvertently remove valid operational data (such as initial measurements in "Pending" status awaiting VEE processing). There is no process for "undoing" the removal of initial measurements and related objects using this feature.

# Understanding Interval Price Set Types

Interval price set types define properties belonging to a class of interval price sets.

Interval price set types are defined by the following:

- **Service Type**: The service type to which interval price sets and interval prices based on this type will be applied
- **Interval Size**: The size of the intervals for interval prices based on this type, represented as hours:minutes:seconds (HH:MI:SS).
- **UOM**: The unit of measure used to define interval price sets and interval prices based on this type
- **TOU**: The time of use period used to define interval price sets and interval prices based on this type
- **SQI**: The service quantity identifier used to define interval price sets and interval prices based on this type
- **Short-Hand Description**: A short-hand description used for interval price sets of this type

> **Note**
>
> Either a UOM or SQI **must** be defined when creating an interval price set type.

### System Data

Interval Price Set Types are based on the following system data:

| Data Type | Description |
|---|---|
| Maintenance Object | Measuring Component Type (D1-MCTYPE) |
| Business Object(s) | Interval Price Set Type (D1-IntervalPriceType) |


Copyright © 2026, Oracle and/or its affiliates.

# Defining Interval Price Set Types

You use the **Interval Price Set Type** portal to display and maintain interval price set types.

Refer to **Understanding Interval Price Set Types** for more information./

You can access the portal by selecting **Admin**, select **Interval Price Set**, then select **Interval Price Set Type**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Interval Price Set Type**: This zone displays basic information about interval price set type.

# Understanding IMD Types

IMD Types are configuration records used to define how initial measurements are processed using Direct Measurement Processing.

There are two main categories of IMD types - seeder types (based on the Seeder IMD Type (D1-IMDSeederType) business object) and specific types (based on the Specific IMD Type (D1-IMDType) business object).

- **Seeder IMD types** represents an initial measurement data (IMD) seeder, which serves as the universal input into the system for external callers or for initial measurement payload processing. The main purpose of the seeder IMD type is to determine the most appropriate specific IMD type (such as interval or scalar, or a type used with a particular Smart Grid Gateway adapter) to use for further processing and prepare the input data for that IMD type. Additionally, the seeder IMD type is used to perform critical validation (such as an interval count validation), device and channel identification, mapping of external values (such as Unit of Measure and time zone) to internal values, and time zone conversion for date/time fields.

  **Note**: There can only be one seeder IMD type active at any one time.

- **Specific IMD types** define the rules for processing a specific initial measurement data (IMD) record. Different specific IMD types contain rules for processing the different types of initial measurement data that can be received. For example, one IMD type defines the rules for processing a scalar measurement from the Landys+Gyr head end system while another IMD Type instance defines the rules for processing an interval initial measurement from the Sensus head end system. There are four types of algorithms supported for specific IMD types: Pre Processing, Pre-VEE, VEE, and Post-VEE. These algorithms are responsible for validating that the measurement has been correctly mapped to the IMD Type (Pre Processing), calculating consumption (Pre-VEE), performing VEE (VEE), writing to the final measurement table (Post-VEE), among other responsibilities.

IMD Types are defined for specific measuring component types in the the "Initial Measurement Creation" processing method (“How to Create MC Related Information”) for each head end system. See Creating Head End Systems for more information.

IMD Types are defined by the following:

- **Initial Measurement Data Seeder**: A flag that indicates (Yes or No) if the IMD type represents a seeder IMD type. There can only be one active IMD type designated as a seeder IMD type at the same time. Seeder IMD types only use Pre-processing Algorithms (see below).


Copyright © 2026, Oracle and/or its affiliates.

- **IMD Business Object**: The business object to use when creating IMDs for the IMD type.
  - For seeder IMD types, this should be "IMD Seeder" (D1-IMDSeeder).
  - For non-seeder IMD types, this should be an initial load IMD business object such as "Initial Load IMD (Interval)" (D1-InitialLoadIMDInterval) or "Initial Load IMD (Scalar)" (D1-InitialLoadIMDScalar).
- **Status**: The status of the IMD type.
- **IMD Information Exception Header BO**: The bunsess to use when creating header records for "Information" exceptions. See VEE Processing and Exceptions with Direct Measurement Processing for more information.
- **Algorithms**: Algorithms used when processing initial measurement data for the IMD type, including:
  - Pre-processing Algorithms
  - Pre-VEE Algorithms
  - VEE Algorithms
  - Post-VEE Algorithms

### System Data

IMD Types are based on the following system data:

| Data Type | Description |
|---|---|
| Maintenance Object | Initial Measurement Data Type (D1-IMDTYPE) |
| Business Object(s) | Specific IMD Type (D1-IMDType) |
| | Seeder IMD Seeder Type (D1-IMDSeederType) |

### Base Package IMD Types

The base package includes several pre-defined IMD types that can be used when processing data from different head end systems supported through Smart Grid Gateway Adapters. These IMD types use vendor-specific initial measurement business objects and algorithms when processing data. The table below lists the IMD types provided with the base package.

| IMD Type | Description |
|---|---|
| IMD Seeder Type (D1-IMDSEEDERTYPE) | Used as the seeder, which determines the appropriate specific IMD type to use and to prepare the input data for that IMD type |
| Interval IMD Type (D1-INT-IMDTYPE) | Used when processing interval data |
| Interval Passthrough IMD Type D1-INT-PT-IMDTYPE) | Used with "pass through" interval external measuring components to capture errors in pass through interval measurements |
| Scalar IMD Type (D1-SCA-IMDTYPE) | Used when processing scalar data |
| Scalar Passthrough IMD Type (D1-INT-SC-IMDTYPE) | Used with "pass through" scalar external measuring components to capture errors in pass through scalar measurements |
| Landis+Gyr - Interval IMD Type (D3-INT-IMDTYPE) | Used when processing interval data using the Landis+Gyr Adapter |
| Landis+Gyr - Scalar IMD Type (D3-SCA-IMDTYPE) | Used when processing scalar data using the Landis+Gyr Adapter |
| MV90 - Interval IMD Type (D5-INT-IMDTYPE) | Used when processing interval data using the MV90 Adapter |


Copyright © 2026, Oracle and/or its affiliates.

| | |
|---|---|
| MV90 - Scalar IMD Type (D5-SCA-IMDTYPE) | Used when processing scalar data using the MV90 Adapter |
| Sensus - Interval IMD Type (D6-INT-IMDTYPE) | Used when processing interval data using the Sensus Adapter |
| Sensus - Scalar IMD Type (D6-SCA-IMDTYPE) | Used when processing scalar data using the Sensus Adapter |
| SSN - Interval IMD Type (D7-INT-IMDTYPE) | Used when processing interval data using the Silver Spring Networks Adapter |
| SSN - Scalar IMD Type (D7-SCA-IMDTYPE) | Used when processing scalar data using the Silver Spring Networks Adapter |
| Itron - Interval IMD Type (D8-INT-IMDTYPE) | Used when processing interval data using the Itron Adapter |
| Itron - Scalar IMD Type (D8-SCA-IMDTYPE) | Used when processing scalar data using the Itron Adapter |
| ADK - Interval IMD Type (DG-INT-IMDTYPE) | Used when processing interval data using the Adapter Development Kit |
| ADK - Scalar IMD Type (DG-SCA-IMDTYPE) | Used when processing scalar data using the Adapter Development Kit |

#### Changing and Extending Base Package IMD Types

The base package IMD types can be changed and/or extended by adding new algorithms to one or more of the algorithm areas (Pre-Processing, Pre-VEE, VEE, or Post-Vee).

When adding new algorithms to a base package IMD type, we recommend creating the algorithm logic in a Groovy Library Script that can be called from an IMD type algorithm as well as from an IMD business object algorithm. See Groovy Library Scripts for more information.

## Configuring IMD Types

You use the **IMD Type** portal to display and maintain IMD types.

Refer to Understanding IMD Types for more information.

You can access the portal by selecting **Admin**, then **Device**, then **IMD Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **IMD Type List**: This zone lists all division records. Broadcast a record to display the details of the selected record.
- **IMD Type**: This zone provides information about the selected division.

# Defining Validation, Editing, and Estimation (VEE) Options

- Exception Types
- VEE Groups
- VEE Rules
- Measuring Component Statistics


Copyright © 2026, Oracle and/or its affiliates.

# Exception Types


- Understanding Exception Types
- Configuring Exception Types


## Understanding Exception Types

Exception Types define the groupings of exceptions for an IMD based on their functional similarity. This provides a way to define VEE Exceptions in a distinct enough way to understand the root issue that was generated from the VEE Rule.

For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

## Configuring Exception Types

This portal is used to display and maintain an Exception Type.

Refer to Understanding Exception Types for more information.

You can access the portal by selecting **Admin**, then **VEE**, then **Exception Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Exception Type List**: displays all of the Exception Types so you can choose the one you want to display in more detail
- **Exception Type**: shows the specific configuration for the selected Exception Type

There are two different options to use when creating an Exception Type:

| Name | Details | Business Object |
|---|---|---|
| VEE Exception | This will create a "normal" VEE Exception that is attached to an IMD for tracking of conditions triggered in the VEE process. | D1-VEEException |
| VEE Exception - Monitor Service Point | In addition to tracking the failure in the VEE process, this VEE Exception generates a Service Issue Monitor. This allows for cumulative tracking of VEE Exceptions that can be configured to result in a Service Investigative Order (field work) for the Service Point. | D2-VEEExceptionServiceMonitor |

# VEE Groups


- Understanding VEE Groups
- Configuring VEE Groups




Copyright © 2026, Oracle and/or its affiliates.

# Understanding VEE Groups

VEE groups are collections of VEE Rules that are applied to initial measurement data. During the VEE process, the system executes the VEE Rules defined in each VEE group. The rules within a VEE group are defined in a specific sequence, allowing control over the order in which the rules are executed.

VEE groups can be associated to a specific measuring component, or to a measuring component type (or both). VEE groups associated with a measuring component type are applied to all measuring components of that type, while those associated to a specific measuring component are applied only to that measuring component. VEE groups associated to a measuring component override those assigned to a measuring component type.

VEE groups can also be referenced by the Execute VEE Group VEE Rule.

For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

**Post-ML VEE Groups**

When using Machine Learning Anomaly Scoring, you can define a VEE Group that is executed after the machine learning processing. These are used to execute rules that compare or adjust the measurement values based on a related measuring component. These groups are referred to as "Post-ML" VEE groups, and are defined for the "Post-ML" VEE Role on measuring component types and measuring components.

Post-ML VEE Groups can only include the following VEE Rules configured for Post-ML:

- Sum Check
- Interval Adjustment From Scalar
- Scalar Calculation from Interval
- Final Measurement Replacement

> **Note**
>
> Post ML VEE Groups can NOT reference Execute VEE Group or VEE Group Matrix rules.

# Configuring VEE Groups

This portal is used to display and maintain a VEE Group.

Refer to Understanding VEE Groups for more information.

You can access the portal by selecting **Admin**, then **VEE**, then **VEE Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **VEE Group**: Defines basic information about VEE group
- **VEE Rules List**: lists the VEE Rules belonging to the group


Copyright © 2026, Oracle and/or its affiliates.

- **Referencing VEE Rules List**: lists the VEE Rules that reference the group
- **Referencing VEE Group Factors List**: lists the VEE group factors that reference the group
- **Referencing Measuring Component Type List**: lists the measuring component types that reference the group
- **Referencing Measuring Component List**: lists the measuring components that reference the group

# VEE Rules


- Understanding VEE Rules
- Configuring VEE Rules


## Understanding VEE Rules

VEE Rules are standard and custom Validation and Estimation rules that perform checking and/or manipulation of initial measurement data. VEE Rules are created for a specific VEE group. For example, if you were configuring two VEE groups and both included a specific VEE Rule, you would need to create two instances of the VEE Rule, one for each group.

The specific validation and estimation processing performed on initial measurement data is defined in individual VEE Rules, each performing a specific set of targeted logic. The base product contains many VEE Rules you can use in your implementation, but you can also create your own custom VEE Rules.

Some VEE Rules generate VEE Exceptions if the initial measurement data fails the conditions specified for the rule. Other rules override measurements, changing measurement values as dictated by the rule's parameters. Some rules can both create exceptions and override the measurement as part of a single process. By convention, VEE Rules change the Post-VEE quantities of initial measurement data, but VEE Rules can change anything on an initial measurement.

Every VEE Rule has an effective period. Rules will only be applied if the initial measurement's start date is within the rule's effective period. For example, an Interval Spike Check rule with a Start Date of 11/15/2010 will only be applied if the start date of the initial measurement is on or after 11/15/2010.

This allows you to update the specifics of a rule without removing the previous version of the rule. For example, you might change the tolerance of an Interval Spike Check rule from 1.2 to 1.5 as of a certain date. However, for initial measurement data for the period prior to the change, you would want to use the tolerance for the original version of the rule (1.2) instead of the new tolerance (1.5).

On almost every VEE Rule, the failure of the rule results in a VEE Exception and the Exception Type for the failure can be configured on the rule. These Exception Types can also be set to a specific Exception Severity:

- **Information**: Used to highlight minor issues, but not sufficient to cause the initial measurement data to be put into the exception state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the initial measurement data from being finalized. Multiple "issue exceptions" can be created during VEE processing. If at least one



Copyright © 2026, Oracle and/or its affiliates.

issue exists after all rules have been applied, the initial measurement data is transitioned to the exception state

- **Terminate**: Used to report a severe issue that will cause the VEE process to stop and the initial measurement data to be transitioned immediately to the exception state. Only one terminate exception can be issued (as the first one causes VEE processing to stop on an initial measurement data).

For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

# Configuring VEE Rules

This portal is used to display and maintain a VEE Rule. Also, a list of the specific out-of-the-box rules is included below the instructions for using the portal.

Refer to Understanding VEE Rules, Understanding VEE Groups, and Understanding Exception Types for more information. For a deeper functional understanding of VEE, refer to the About VEE section of the *Oracle Utilities Meter Solution Business User Guide*.

You can access the portal by selecting **Admin**, then **VEE**, then VEE Rule. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **VEE Rule**: this zone displays all of the configuration items specific to this instance of a VEE Rule.
  - Click **Show All Rules** to open the **VEE Rules** List zone, which lists all of the VEE rules in the VEE group to which the current rule belongs.
- **Eligibility Criteria List**: this zone displays any VEE Rule Eligibility Criteria that have been setup. These eligibility criteria determine conditionally whether the VEE Rule should be executed or not. Use the Add button to create a new eligibility criteria for the rule you're viewing.

### Validation VEE Rules

Below is a list of the validation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| VEE Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Consecutive Interval Check | Interval | This validation rule flags any combination of consecutive intervals within Initial Measurement Data based on the values of the data or the condition codes of the data. |
| Duplicate IMD Check | Interval or Scalar | This rule allows for a duplicate IMD to be flagged. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Dynamic Comparison Validation | Interval or Scalar | This powerful, flexible validation rule compares measurements to historical statistics for the related Service Point. The system will maintain statistics such as the following: sum, min, max, average, median, zero value count, outage count, and standard deviation. Then you define formulas (no programming required) for the comparison of current measurements to these statistics. |
| Ensure IMD Exists for Sibling MCs | Interval or Scalar | This rule validates that an IMD exists for all of the other measuring components associated to the same Device Configuration as the current measuring component, for the same period of time. |
| Final Measurement Replacement | Interval or Scalar | This validation rule allows you to define a variety of configuration options to decide if new data should replace existing measurements. The options include value change thresholds, percentage change thresholds, as well as condition code ranking. |
| High/Low Check | Interval or Scalar | This validation rule compares the total consumption of the current IMD to historical values. The comparison is normalized based on average daily usage (ADU). If the current IMD is too high or too low compared to historical data then an exception is thrown. |
| Inactive Measurement Check | Interval or Scalar | This validation rule flags any Initial Measurement Data received on a device that is either disconnected, uninstalled, and/or not connected to a Usage Subscription. |
| Interval Size Validation | Interval | This rule validates that the interval size (in seconds) supplied with the Initial Measurement is equal to the interval size defined on measuring component's type. |
| Interval Spike Check | Interval | This rule checks for spikes within an IMD and generates an exception if one is found. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Multiplier Check | Interval or Scalar | This rule validates that the register multiplier supplied with the IMD is equal to the multiplier stored on the measuring component. If not, an exception is created using the register multiplier exception type and severity configured on the rule. |
| Negative Consumption Check | Interval or Scalar | This rule flags any IMD where the total consumption is negative. |
| Prolonged Estimation Check | Interval or Scalar | This validation rule creates an alert when a device has been estimated for an extended period of time. |
| Raise Missing Quantity Exception | Interval | This rule flags any missing interval data. |
| Sum Check | Interval or Scalar | This rule is used to compare the difference between interval data to scalar data for a period of time, or between a set of TOU scalar reads to a "Sum" scalar reading. |
| Unit of Measure Check | Interval or Scalar | This rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type. |
| Zero Consumption Check | Interval or Scalar | This rule checks if the total consumption for the IMD is zero. There is also a check for whether an outage occurred during the same time as the zero consumption to provide ways to avoid exceptions in that case. |

### Estimation VEE Rules

Below is a list of the estimation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| VEE Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Interval Adjustment From Scalar | Interval | This rule performs adjustments to interval values based on the values from the associated scalar data on the same device. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Interval Averaging Estimation | Interval | This rule finds historical interval data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD. |
| Interval Create Estimation IMD for Gap | Interval | This estimation rule creates a new Estimation IMD to fill gaps between the interval data received from the Head End.<br>***Note:*** *this estimation rule acts differently from the other rules as it creates a new IMD rather than filling in missing values in an estimation IMD created from periodic estimations.* |
| Interval Interpolation Estimation | Interval | This rule attempts to interpolate gaps within an IMD using prior and subsequent intervals as starting points for linear interpolation. |
| Interval Profile Estimation | Interval | This rule estimates any missing interval data for the IMD based on a referenced Profile Factor. |
| Interval Proxy Day Estimation | Interval | This rule checks for days with similar temperature and uses the measurement data from those days as the basis of interval estimation for the in-flight initial measurement. |
| Scalar Calculation From Interval | Scalar | This rule performs adjustments to scalar values based on the values from the associated interval data on the same device. |
| Scalar Estimation | Scalar | This rule finds historical scalar data from the same measuring component based on a variety of configuration options to use for estimating any missing data within an IMD. |
| Scalar Profile Estimation | Scalar | This rule estimates any missing scalar data for the IMD based on a referenced Profile Factor. |
| Scalar Proration | Scalar | This rule prorates the value of a scalar reading that has two valid scalar readings on either side as boundaries. It will also take into account any related interval data within the same period to exclude from the calculation. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Subtractive Interval Adjustment Rule | Subtractive Interval Adjustment Rule | This rule performs adjustments to qualifying interval consumption values for subtractive interval measuring components based on an adjustment target calculated using a start and stop reading for the period that encapsulates the intervals to adjust. |

### Decision-Making VEE Rules

There are VEE Rules delivered as part of the base product that help with decision-making when executing VEE (listed below). For more information on how each rule executes and can be configured, follow the link provided on the rule.

| VEE Rule Name | Purpose |
|---|---|
| Exception Handler | This rule allows for termination of the VEE process based on a configurable set of exceptions being present for the IMD. This rule also allows a unique To Do Type to be generated based on a group of exceptions. |
| Execute VEE Group | This rule performs a call to execute a separate VEE Group which includes execution of all VEE Rules within that group. |
| Successful Termination | This rule allows VEE to be successfully terminated based on a list of exceptions. |
| VEE Group Matrix (Factor) | This rule provides a way to choose different instances of a VEE Rule using a Factor. This factor leverages characteristics that are defined on Service Point, Device, or Measuring Component. |

- Validation VEE Rules
- Estimation VEE Rules
- Decision-Making VEE Rules

## Validation VEE Rules

- Consecutive Interval Check
- Duplicate IMD Check
- Dynamic Comparison Validation
- Ensure IMD Exists for Sibling MCs
- Final Measurement Replacement
- High/Low Check
- Inactive Measurement Check
- Interval Size Validation
- Interval Spike Check


Copyright © 2026, Oracle and/or its affiliates.

- Multiplier Check
- Negative Consumption Check
- Prolonged Estimation Check
- Raise Missing Quantity Exception
- Sum Check
- Unit of Measure Check
- Zero Consumption Check

## Consecutive Interval Check

### Overview

This rule flags any combination of consecutive intervals within Initial Measurement Data based on the values of the data or the condition codes of the data. It can be used to find faulty meters that are reporting consecutive outage codes, zero measurements, or negative values. It can also be used by water utilities to identify leaks based on the interval never reaching zero.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-CONSINTRV Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

### Business Object

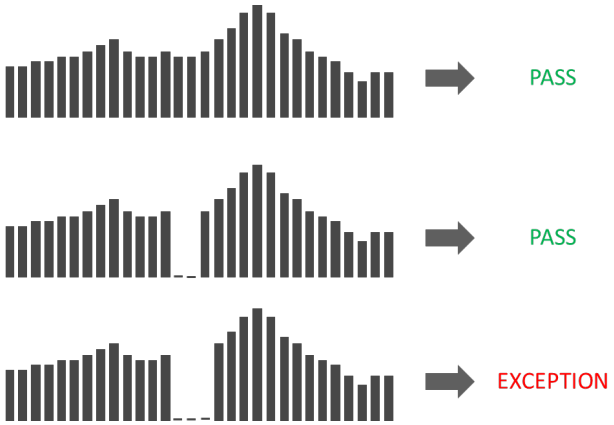
D2-ConsecutiveIntervalCheck

### Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.

**Scenario 1:** Rule configured to fail for zero values

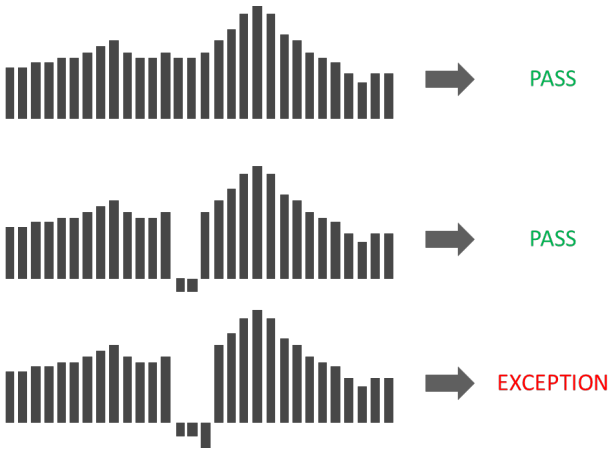
| Parameter Name | Configuration Value |
|---|---|
| Check performed | Consecutive Measurement Values |
| Maximum consecutive hours | 02:00:00 |
| Look To Prior Measurements | No |
| Equation Type | Equal To |
| Value for Comparison | 0 |
| Bottom Range Condition | |
| Top Range Condition | |




**Scenario 2:** Rule configured to fail for negative values


Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Configuration Value |
|---|---|
| Check performed | Consecutive Measurement Values |
| Maximum consecutive hours | 02:00:00 |
| Look To Prior Measurements | No |
| Equation Type | Less Than |
| Value for Comparison | 0 |
| Bottom Range Condition | |
| Top Range Condition | |




**Scenario 3**: Rule configured to fail for missing condition codes

| Parameter Name | Configuration Value |
|---|---|
| Check performed | Consecutive Condition Values |
| Maximum consecutive hours | 02:00:00 |
| Look To Prior Measurements | No |
| Equation Type | |
| Value for Comparison | |
| Bottom Range Condition | Missing |
| Top Range Condition | Missing |

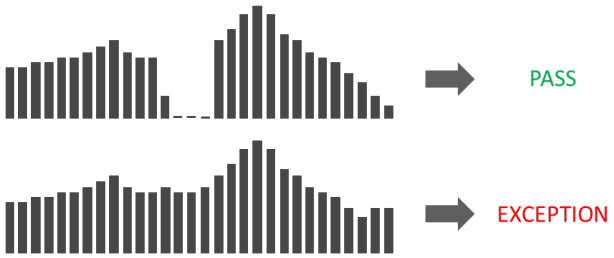
**Scenario 4**: Rule configured to fail for outage condition codes only if its zero

| Parameter Name | Configuration Value |
|---|---|
| Check performed | Both |
| Maximum consecutive hours | 02:00:00 |
| Look To Prior Measurements | No |
| Equation Type | Equal To |
| Value for Comparison | 0 |
| Bottom Range Condition | Outage |
| Top Range Condition | Outage |

**Scenario 5**: Rule configured to fail if a long period of consecutive water values are greater than 0.1

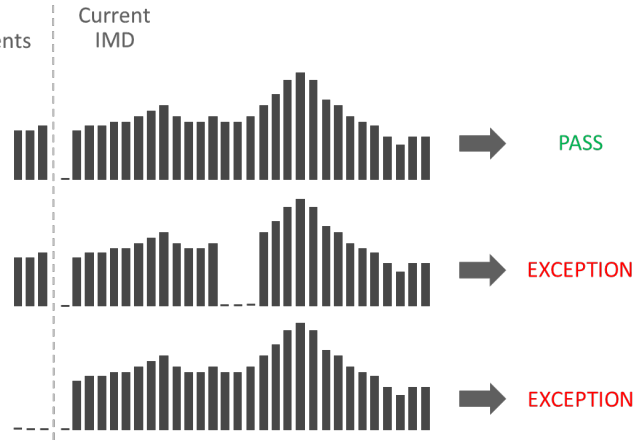

Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Configuration Value |
|---|---|
| Check performed | Consecutive Measurement Values |
| Maximum consecutive hours | 20:00:00 |
| Look To Prior Measurements When First Interval Matches Criteria | No |
| Equation Type | Greater Than |
| Value for Comparison | 0.1 |
| Bottom Range Condition | |
| Top Range Condition | |




**Scenario 6**: Look back at prior measurements when the first interval matches criteria

| Parameter Name | Configuration Value |
|---|---|
| Check performed | Consecutive Measurement Values |
| Maximum consecutive hours | 02:00:00 |
| Look To Prior Measurements | Yes |
| Equation Type | Equal To |
| Value for Comparison | 0 |
| Bottom Range Condition | |
| Top Range Condition | |




## Duplicate IMD Check

If the current Initial Measurement Data (IMD) being validated is determined to be a duplicate of an existing IMD for the same measuring component, this rule will produce a VEE exception of the type and severity configured on the VEE Rule. The algorithm logic looks for duplicate IMDs using the following criteria:

- associated to the same measuring component as the current IMD
- utilizes the same IMD business object as the current IMD (for example, Initial Load Scalar)
- references the same To Date/Time (ends on the same date) as the current IMD
- exists in a "Finalized" state
- the contents of pre-VEE are identical to the pre-VEE of the current IMD

If any IMDs are found that meet all of the above criteria, the current IMD is deemed to be a duplicate.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-DUPIMDCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.


Copyright © 2026, Oracle and/or its affiliates.

**Business Object**

D1-DuplicateIMDCheck

## Dynamic Comparison Validation

This validation rule compares measurements to historical statistics for the related Service Point. The system will maintain statistics such as the following: sum, min, max, average, median, zero value count, outage count, and standard deviation. Setting up these Measuring Component Statistics is a prerequisite to using the rule.

Users can define formulas (no programming required) for the comparison of current measurements to these statistics. This is powerful rule will allow utilities to look for unusual usage patterns. For example:

- Lowest/highest usage ever
- Current usage is more than three standard deviations from the mean
- Detect unexpected zero usage
- Detect negative usage while ruling out known cases
- Abnormal usage
- Voltage threshold monitoring
-

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-DYNCOMCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

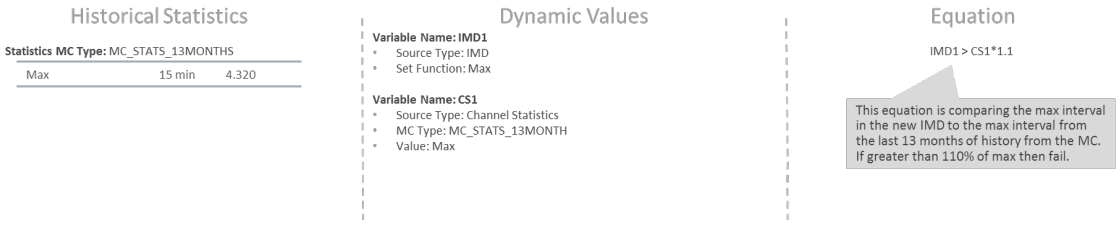
**Business Object**

D2-DynCompValidation

**Example Scenarios**

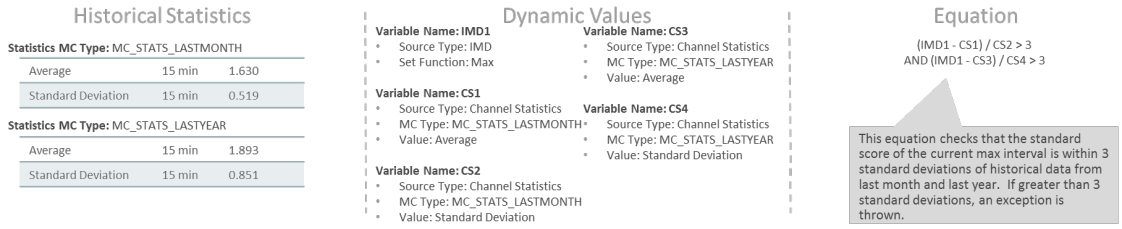
Below are some example scenarios that can be achieved based on configuration of this rule.

**Example 1:** Detect new "high water mark"

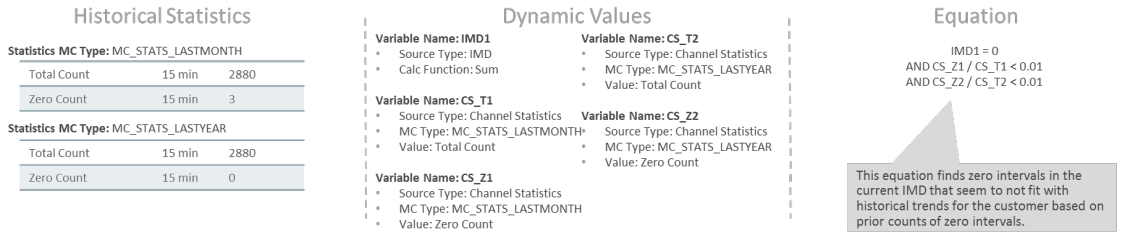



**Example 2:** High Usage

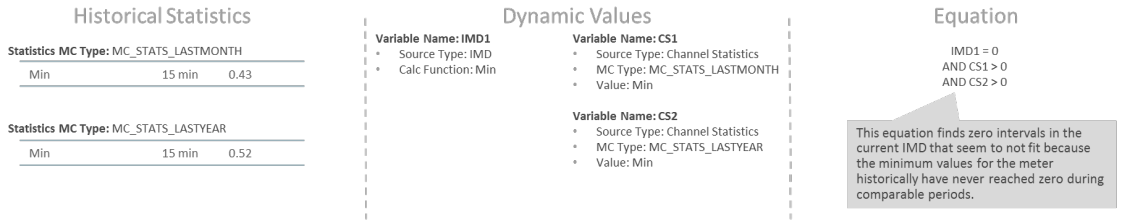

Copyright © 2026, Oracle and/or its affiliates.



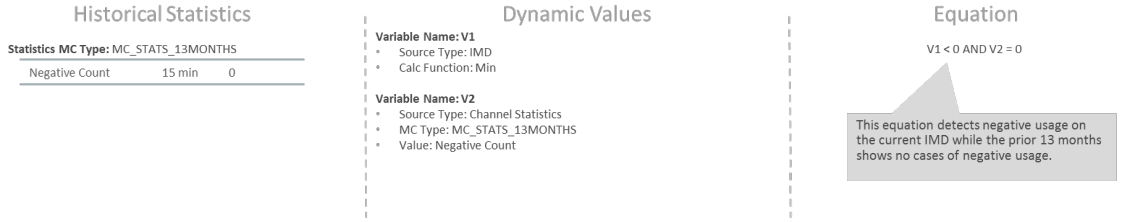

**Example 3:** Detect unexpected zero usage (method 1)



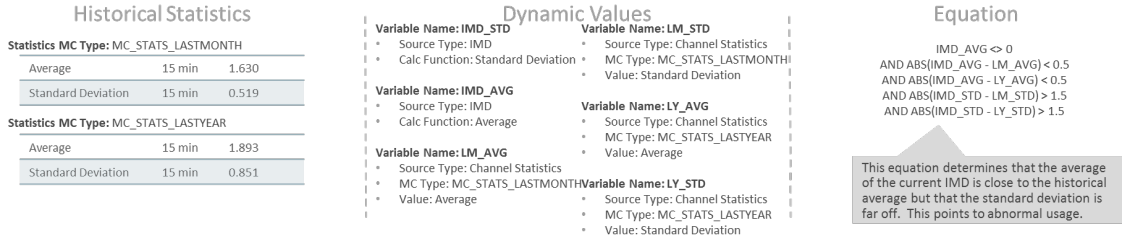

**Example 4:** Detect unexpected zero usage (method 2)




**Example 5:** Detect negative usage while ruling out known cases

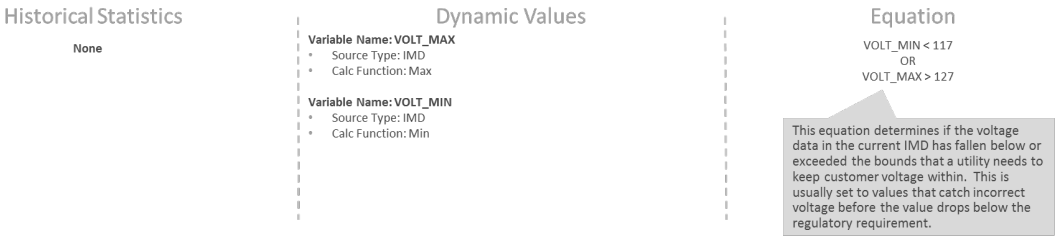



**Example 6:** Abnormal usage




**Example 7:** Voltage threshold monitoring


Copyright © 2026, Oracle and/or its affiliates.




## Ensure IMD Exists for Sibling MCs

This rule validates that an IMD exist for all of the other measuring components associated to the same Device Configuration as the current measuring component, for the same period of time.

A check is also performed that all of the Initial Measurements have the same External Source ID (indicating that they all came from the same usage file).

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ENSIMDMC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-EnsureIMDExistsForSibling

## Final Measurement Replacement

This validation rule allows you to define a variety of configuration options to decide if new scalar or interval data should replace existing measurements. The options include value change thresholds, percentage change thresholds, as well as condition code ranking. One common use for this rule is rejecting trivial measurement changes to prevent very small changes for a bill from being sent to a customer.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-VLMSRCOND Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

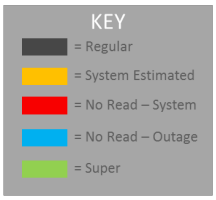
D2-FinalMeasurementValidation

**Example Scenarios**

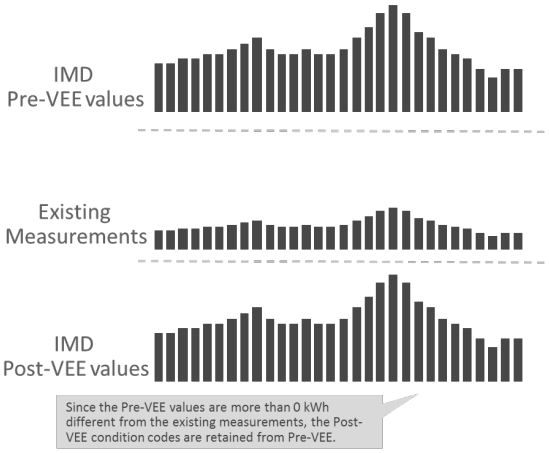

Copyright © 2026, Oracle and/or its affiliates.

Below are some example scenarios that can be achieved based on configuration of this rule.

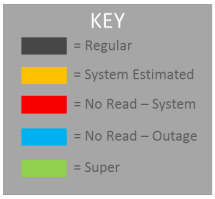
**Example 1:** Replacement of measurements based solely on value difference



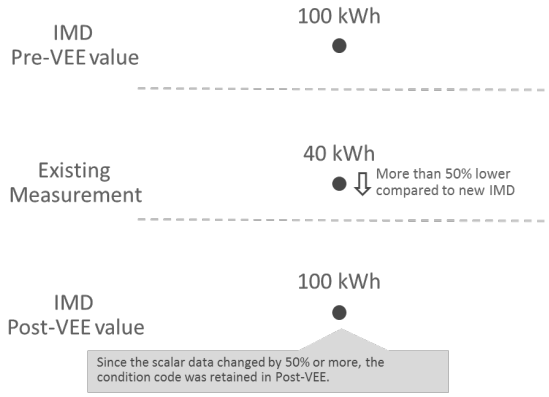

| Parameter Name | Configuration Value |
|---|---|
| Replacement Method | Reject Based on Configuration |
| Replacement Condition Value | No Read – System |
| Replace Manually Edited Data? | |
| Use Early and Late MC Type Settings? | |
| Replacement Evaluation Type | Compare Value Change |
| Value Change Tolerance | 0.1 |
| Percentage Change Tolerance | |
| Condition Code Prioritization by Groups | |




**Example 2:** Percentage change to replace measurement

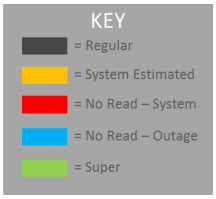



| Parameter Name | Configuration Value |
|---|---|
| Replacement Method | Reject Based on Configuration |
| Replacement Condition Value | No Read – System |
| Replace Manually Edited Data? | |
| Use Early and Late MC Type Settings? | |
| Replacement Evaluation Type | Compare Percentage Change |
| Value Change Tolerance | |
| Percentage Change Tolerance | 50 |
| Condition Code Prioritization by Groups | |

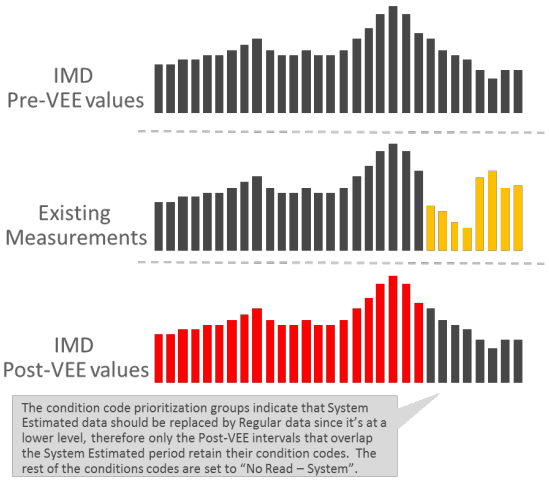



**Example 3:** Partial replacement of estimated data with regular data based on condition range prioritization

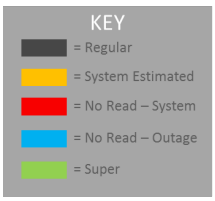

Copyright © 2026, Oracle and/or its affiliates.



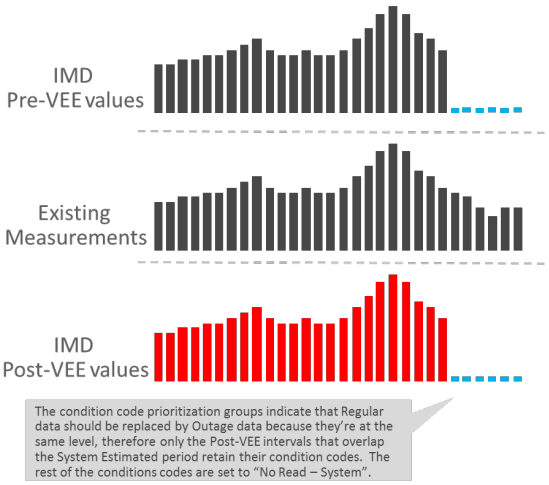

| Parameter Name | Configuration Value |
|---|---|
| Replacement Method | Reject Based on Configuration |
| Replacement Condition Value | No Read – System |
| Replace Manually Edited Data? | |
| Use Early and Late MC Type Settings? | |
| Replacement Evaluation Type | Compare Value Change |
| Value Change Tolerance | 0.1 |
| Percentage Change Tolerance | |
| Condition Code Prioritization by Groups | 10: Super<br>20: Regular, No Read – Outage<br>30: System Estimate<br>40: No Read – System |




**Example 4:** Partial replacement of regular data with outage data based on condition range prioritization

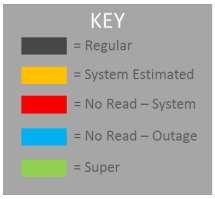



| Parameter Name | Configuration Value |
|---|---|
| Replacement Method | Reject Based on Configuration |
| Replacement Condition Value | No Read – System |
| Replace Manually Edited Data? | |
| Use Early and Late MC Type Settings? | |
| Replacement Evaluation Type | Compare Value Change |
| Value Change Tolerance | 0.1 |
| Percentage Change Tolerance | |
| Condition Code Prioritization by Groups | 10: Super<br>20: Regular, No Read – Outage<br>30: System Estimate<br>40: No Read – System |

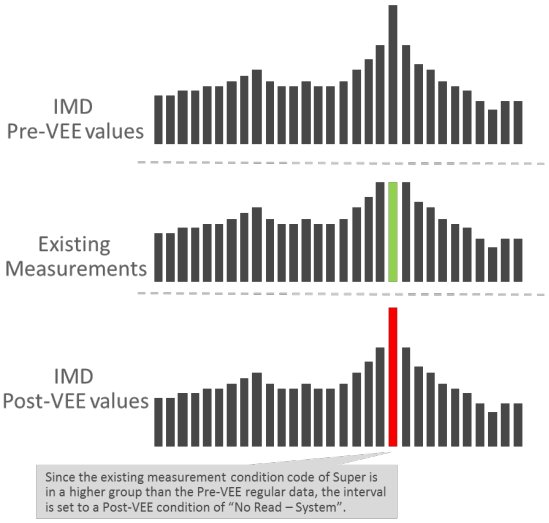



**Example 5:** Super not replaced by regular (retain a smoothed spike)

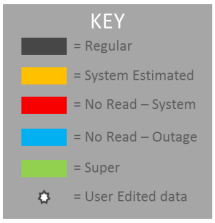

Copyright © 2026, Oracle and/or its affiliates.



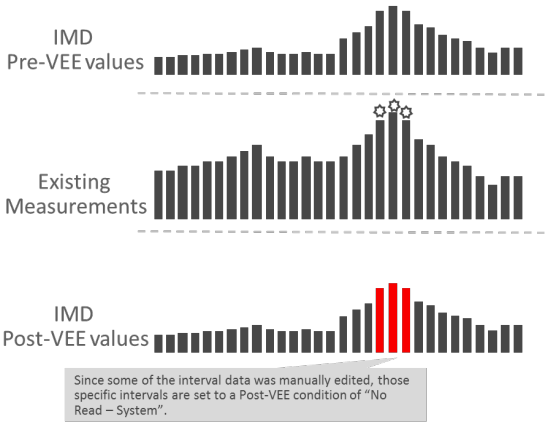

| Parameter Name | Configuration Value |
|---|---|
| Replacement Method | Reject Based on Configuration |
| Replacement Condition Value | No Read – System |
| Replace Manually Edited Data? | |
| Use Early and Late MC Type Settings? | |
| Replacement Evaluation Type | |
| Value Change Tolerance | |
| Percentage Change Tolerance | |
| Condition Code Prioritization by Groups | 10: Super<br>20: Regular, No Read – Outage<br>30: System Estimate<br>40: No Read – System |




**Example 6:** Prevent replacement of manually edited data

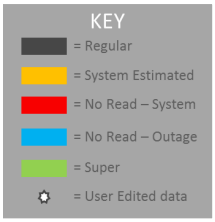



| Parameter Name | Configuration Value |
|---|---|
| Replacement Method | Reject Based on Configuration |
| Replacement Condition Value | No Read – System |
| Replace Manually Edited Data? | No |
| Use Early and Late MC Type Settings? | |
| Replacement Evaluation Type | |
| Value Change Tolerance | |
| Percentage Change Tolerance | |
| Condition Code Prioritization by Groups | |

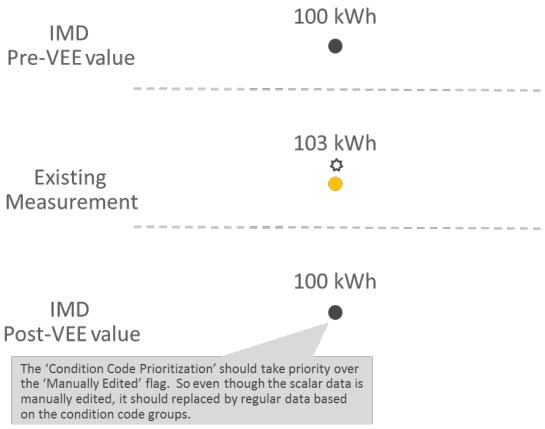



**Example 7:** Allow replacement of manually edited data if the condition group is lower than new data


Copyright © 2026, Oracle and/or its affiliates.




| Parameter Name | Configuration Value |
|---|---|
| Replacement Method | Reject Based on Configuration |
| Replacement Condition Value | No Read – System |
| Replace Manually Edited Data? | No |
| Use Early and Late MC Type Settings? | |
| Replacement Evaluation Type | |
| Value Change Tolerance | |
| Percentage Change Tolerance | |
| Condition Code Prioritization by Groups | 10: Super<br>20: Regular, No Read – Outage<br>30: System Estimate<br>40: No Read – System |




## High/Low Check

This validation rule compares the total consumption of the current IMD to historical values. The comparison is normalized based on average daily usage (ADU). If the current IMD is too high or too low compared to historical data then an exception is thrown.

Numerous configuration options are provided including:

- A Historical Pre-Window and Post-Window that determines the number of days to check before and after the period being examined.
- Percentages to control how much of the historical data can be user edited, estimated, or non-normal data.
- What type of historical data to look at first
- What to do when an outage has occurred

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-HILO-CHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-VEERuleHighLowCheck

## Inactive Measurement Check

This validation rule flags any Initial Measurement Data received on a device that is either disconnected, uninstalled, and/or not connected to a usage subscription.


Copyright © 2026, Oracle and/or its affiliates.

- When the 'Check Measurements on Disconnected Device' flag is set to "Yes", the logic will compare the IMD measurements total to the threshold configured. If the total measurements are above the threshold, then logic compares the IMD dates to two things: 1) On/Off Dates related to the Install Event and 2) the Installation and Removal Dates of the Install Event. If the On or Off falls within the dates for an interval IMD, then the logic will only sum intervals that fall during the disconnected periods.
- When the 'Check for Uninstalled Device' flag is set to "Yes", the logic will check for a valid Install Event based on the IMD dates.
- When the 'Check for Missing Usage Subscription' flag is set to "Yes", the logic will check for an active Usage Subscription based on the IMD dates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INACTVCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

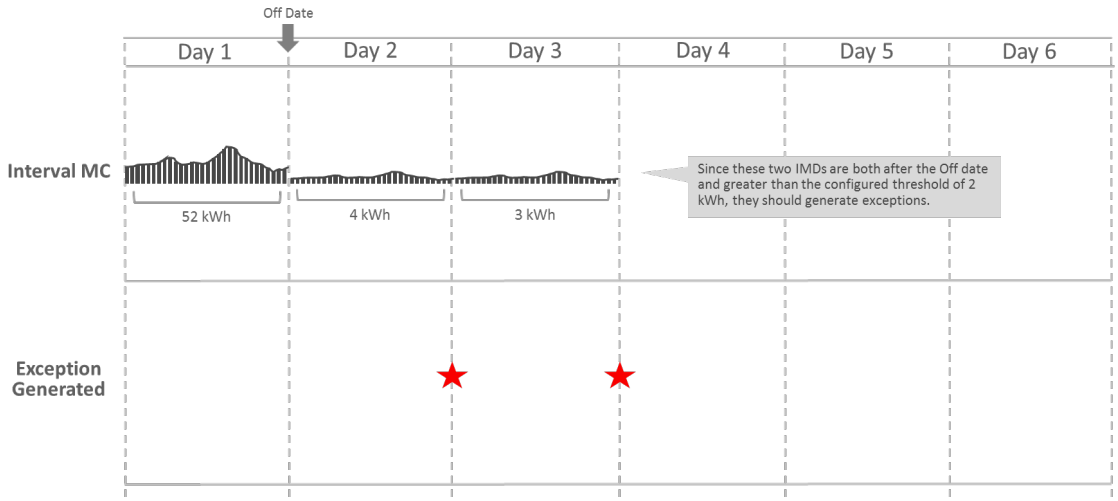
**Business Object**

D2-InactiveMeasurementCheck

**Example Scenarios**

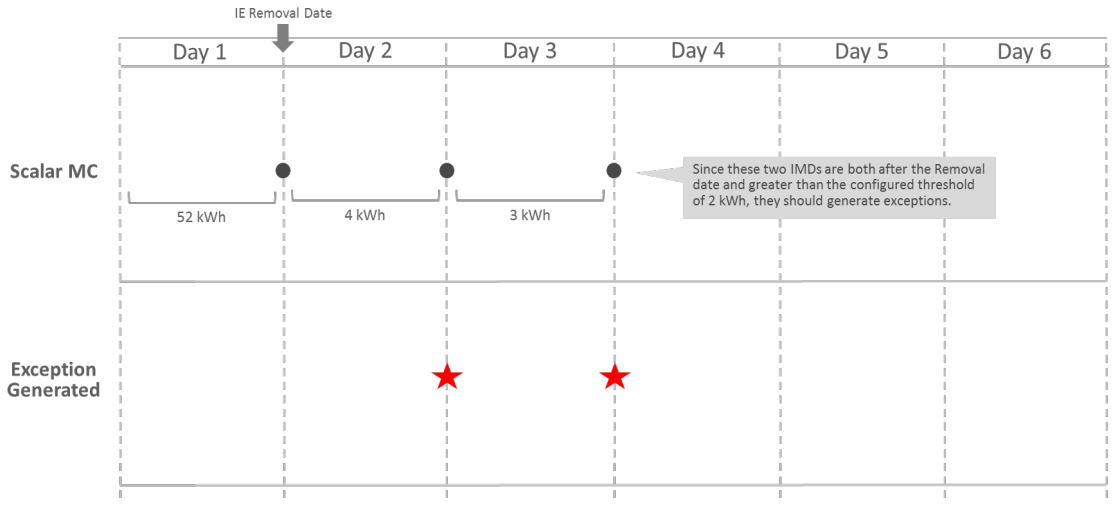
Below are some example scenarios that can be achieved based on configuration of this rule.

**Example 1:** Non-zero data during disconnected period

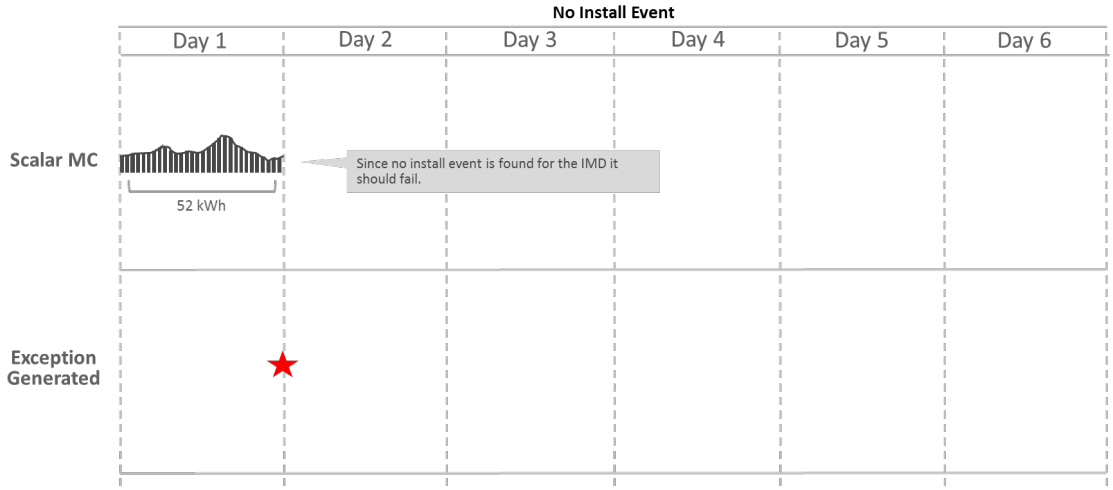



**Example 2:** Non-zero data after removal date with no new install

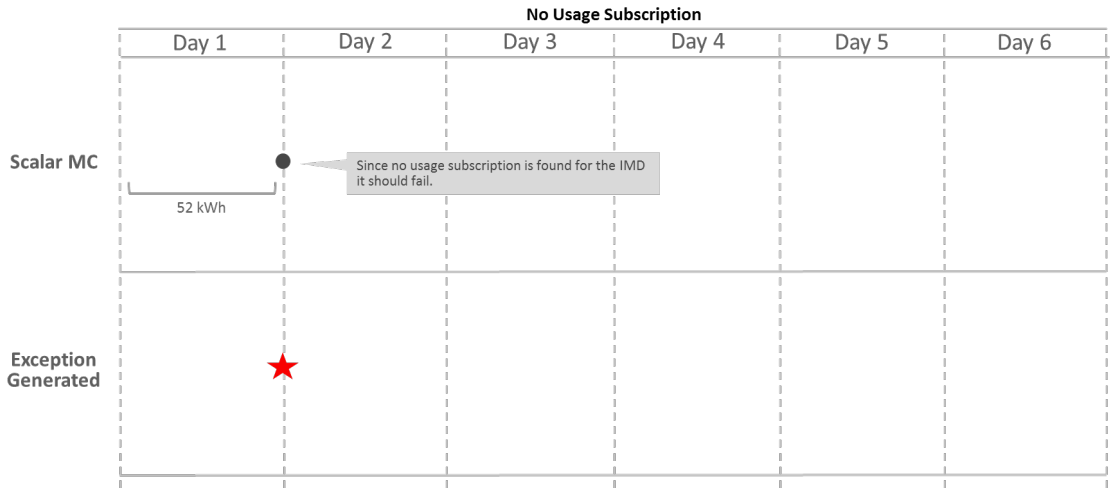

Copyright © 2026, Oracle and/or its affiliates.




**Example 3:** Data received when device has no Install Event records




**Example 4:** Data received when not connected to a Usage Subscription record




## Interval Size Validation


Copyright © 2026, Oracle and/or its affiliates.

This rule validates that the interval size (in seconds) supplied with the Initial Measurement is equal to the interval size defined on measuring component's type.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTSIZVAL Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-IntervalSizeValidation

## Interval Spike Check

This rule looks for spikes by taking the highest interval and the third-highest interval, and determining the percent difference between the two. If the percent difference is larger than the tolerance configured on the rule, the algorithm logs an exception of the type and severity configured on the rule.

The rule can be executed in one of two modes (as configured on the rule):

- The spike check is performed for every 24-hour chunk of data in the initial measurement data set
- The spike check is performed for the entire initial measurement data set

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTSPKCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-IntervalSpikeCheck

## Multiplier Check

This rule validates that the register multiplier supplied with the Initial Measurement is equal to the multiplier stored on the measuring component.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-REGMULCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-RegisterMultiplierCheck

## Negative Consumption Check

This rule checks if the total consumption of the initial measurement is less than zero. If the rule encounters negative consumption, an exception will be logged if the related Measuring Component Type is not configured to "allow negative consumption", or if the Override MC Type Allow Negative Consumption flag on the rule is set to “Yes”.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-NCON-CHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-NegativeConsumptionCheck

## Prolonged Estimation Check

This validation rule can be used on either interval or scalar and will alert you when a device has been estimated for an extended period of time. The IMD must first fall within the condition code range configured on the rule for this to execute. Next the validation finds the 'Most Recent Non-Estimated Read Date Time' from the Measurement Component. If this date plus the 'Days of Estimation Allowable' from the VEE Rule is less than the IMD End Date then a Prolonged Estimation Exception is created.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-PROESTCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

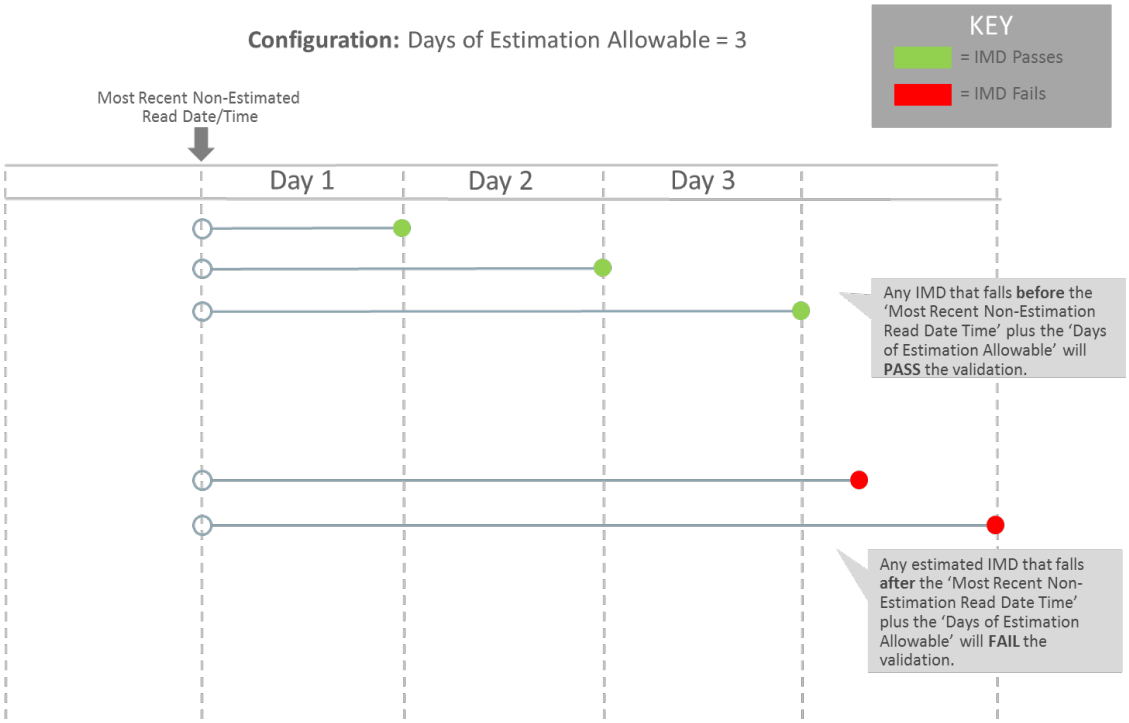
**Business Object**


Copyright © 2026, Oracle and/or its affiliates.

D2-ProlongedEstimationCheck

**Example Scenario**

Below is an example scenario that can be achieved based on configuration of this rule.




## Raise Missing Quantity Exception

This rule flags any missing interval data while providing a soft parameter on the algorithm to exclude a condition range if desired.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-RAIMISQTY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-RaiseMissingQuantityExcp

## Sum Check

This rule evaluates whether consumption for the current Initial Measurement Data is within a tolerance of the sum of the consumption during the same period for any measuring components related to the current one. If the values are not within the defined tolerance of each other, and exception is logged.


Copyright © 2026, Oracle and/or its affiliates.

The rule can be used to evaluate consumption totals for an interval measuring component that has a related scalar measuring component with the same UOM to ensure that the total consumption of the interval measuring component is within a tolerance of that of the scalar value. It can also be used to evaluate consumption totals for scalar TOU meters that have a "check" register (for example, three registers that measure ON-PEAK, OFF-PEAK, and SHOULDER, with a fourth check register that measures the total consumption).

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SUM-CHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-SumCheck

## Unit of Measure Check

This rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-UOMCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-UOMCheck

## Zero Consumption Check

This rule checks if the total consumption for the IMD is zero. There is also a check for whether an outage occurred during the same time as the zero consumption to provide ways to avoid To Dos by having a different exception type.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ZEROCNCHK Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.


Copyright © 2026, Oracle and/or its affiliates.

**Business Object**

D2-ZeroConsumptionCheck

## Estimation VEE Rules

- Interval Adjustment From Scalar
- Interval Averaging Estimation
- Interval Create Estimation IMD for Gap
- Interval Interpolation Estimation
- Interval Profile Estimation
- Interval Proxy Day Estimation
- Scalar Calculation From Interval
- Scalar Estimation
- Scalar Profile Estimation
- Scalar Proration
- Subtractive Interval Adjustment Rule

### Interval Adjustment From Scalar

This rule adjusts interval Initial Measurement Data based on scalar values using one of two options configured on the VEE Rule. Both options require that a scalar measuring component be related to the current interval measuring component using the Consumption Reference Measuring Component, and that one or more final measurements be present for the related measuring component between the start and end date/times of the current Initial Measurement Data set.

1. Adjust all intervals. In this case, the scalar consumption provides a value that is then used to proportionally adjust all of the intervals in the set. The formula used is (Scalar Consumption / Total Initial Measurement Consumption) * Interval Amount. If the total of all of the intervals had been equal to zero originally, the rule adjusts all of the intervals to the same value.
2. Adjust intervals based on condition. For this option, a range of conditions is configured on the VEE Rule, and the rule adjusts only those intervals with conditions that lie within the configured range. The adjustment is done by applying an adjustment factor to each of the intervals within range.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTADJSCA Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-IntervalAdjustmentFrmScalar


Copyright © 2026, Oracle and/or its affiliates.

## Interval Averaging Estimation

The algorithm estimates when the current Measuring Component is eligible. Estimation is performed only if all these conditions are satisfied:

- The Measuring Component is interval.
- The Measuring Component is linked to a Service Point or the VEE Rule's Estimate If Not Attached to SP field is set to "Estimate".
- The Percentage of Missing Intervals is less than the VEE Rule's Maximum Percentage Missing Intervals Threshold.

The algorithm will compute the Total Accumulated Consumption and the Total Number of Intervals using a variety of methods including scanning holidays, similar days of the week, or neighboring days. Once a set of valid comparison days are found then these are used as the basis of creating estimated intervals for the current IMD.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTAVGEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-IntervalAveragingEstimation

## Interval Create Estimation IMD for Gap

This rule is very different from other VEE rules in that it does not examine the current initial measurement but rather looks to see if there are any missing measurements prior to the initial measurement being processed (i.e. a gap exists). If so and the scenario meets the configured options it will generate an estimation initial measurement to fill in that gap.

This rule is intended to provide more real-time filling of missing measurements as opposed to running periodic estimation. However, it is still expected that periodic estimation will be used in conjunction with this rule such that any gaps that are not filled in by this rule would eventually be filled in by periodic estimation.

This rule can be configured to perform minimal validation of that gap that is identified and defer to the estimation initial measurement to validate against other initial measurements and final measurements that may overlap the gap. Conversely, it can also be configured to validate the gap and exclude any periods where a final measurement or in progress initial measurement overlaps the gap's duration.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This estimation rule acts differently from the other rules - it creates a new IMD rather than filling in missing values in an estimation IMD created from periodic estimations.
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-CREESTIMD Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

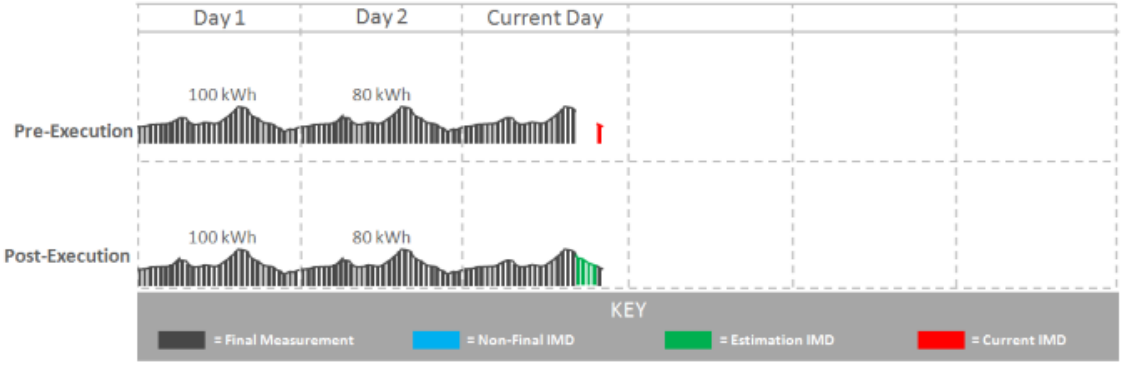
**Business Object**

D2-CreateEstimationIMDRule

**Example Scenarios**

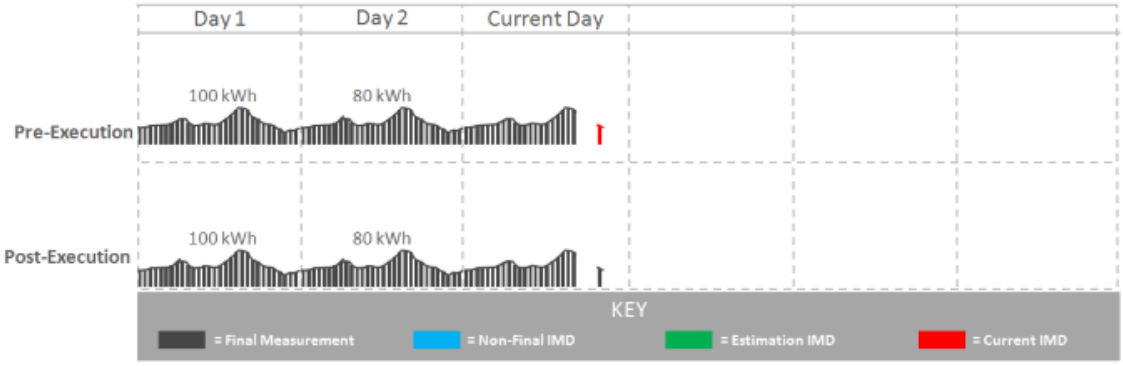
The following are sample scenarios that can be achieved based on configuration of this rule.

**Example 1**: A 4-hour gap exists and the rule is configured to always fill gaps. Maximum hours are set to 4 and the gap fill is set to "always".




Since the maximum hours has not been exceeded and the rule is configured to always fill the gap, a 4-hour estimation IMD is created to fill the gap.

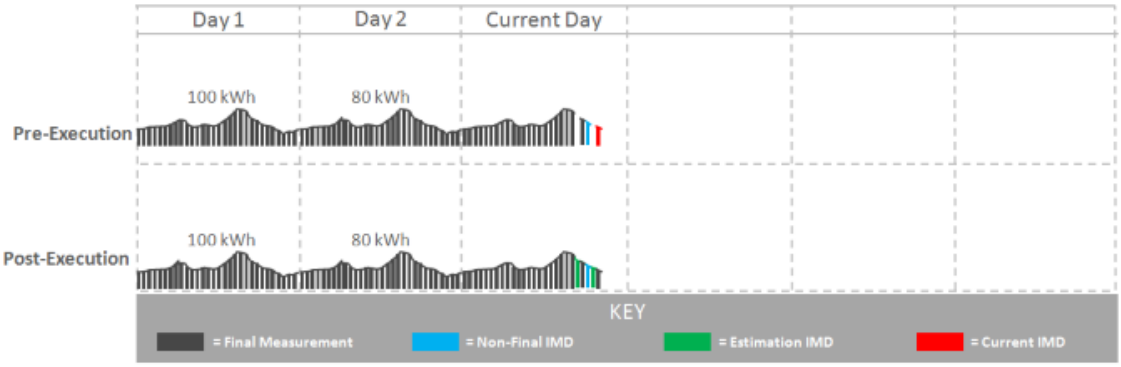
**Example 2**: A 4-hour gap exists and the rule is configured to always fill gaps. However, the maximum number of hours to fill is 3.




Since the gap is 4 hours long and the maximum hours to gap fill is 3 hours, no estimation has been created. An exception would be created if the rule were configured this way.


Copyright © 2026, Oracle and/or its affiliates.

**Example 3**: A non-contiguous gap exists and the rule is configured to "Skip In-Progress IMDs and Final Measurements" with a maximum number of IMDs set to 2.




In this scenario:

- One interval in the 4-hour gap is covered by a final measurement.
- One interval in the 4-hour gap is covered by an in-progress IMD.
- Two estimation IMDs are generated one to fill each hour not covered by a final measurement or in-progress IMD.

## Interval Interpolation Estimation

This rule attempts to interpolate gaps in Initial Measurement Data sets using prior and subsequent intervals as starting points for linear interpolation. A configuration parameter determines the maximum number of consecutive intervals within a gap before that gap can no longer be interpolated.

If a given gap lies at the beginning or end of the Initial Measurement Data set, this algorithm seeks out final measurements immediately before or after the set in an attempt to find two reference measurement values for interpolation. In this situation, assuming that the condition of the measurement retrieved lies in NEITHER of the algorithm parameter ranges you have configured (the measurement is neither "missing" nor an "outage"), this measurement value is then used as the basis for interpolation.

In the event that the gap is the entire length of the Initial Measurement Data set (and the VEE Rule is configured such that this is not too large of a gap), the routine attempts to find final measurements as described above.

If a valid measurement can be found for only one side of a given gap, the interpolation logic assigns each interval in the gap the same value - the value of the measurement retrieved. This is referred to as applying a "flat load".

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTINTEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**


Copyright © 2026, Oracle and/or its affiliates.

D2-IntervalInterpolationEst

## Interval Profile Estimation

This rule uses a profile measuring component's interval consumption as a source of values to assign to intervals in the current Initial Measurement Data set that are marked "missing" via the interval's condition. If the interval is marked with a condition indicating "outage", the algorithm skips it.

If a measurement is not available for the profile measuring component on the date/time, the interval is left unchanged.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTPROEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-IntervalProfileEstimation

## Interval Proxy Day Estimation

This rule checks for days with similar temperature and uses the measurement data from those days as the basis of interval estimation for the in-flight initial measurement. Measuring component comparison periods are used as the main driver for determining similar temperature days.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-INTPROXY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-IntervalProxyDayEstimation

## Scalar Calculation From Interval

This rule calculates a single consumption amount for a scalar measuring component's Initial Measurement using the total consumption for the same date/time range for a related interval measuring component.

The scalar value replaces any existing value within the Initial Measurement (in the post-VEE list) and updates the condition to the value configured on the VEE Rule. This VEE Rule


Copyright © 2026, Oracle and/or its affiliates.

updates the condition to the value defined in "Condition Value for High Quality Condition" when the measurement condition value for all underlying interval data and the previous scalar measurement (when Evaluate Condition Of Previous Scalar Measurement is configured as "Yes") is greater than or equal to the "Minimum Condition Quality to Override" (If "Minimum Condition Quality to Override" is defined) otherwise it updates the measurement condition of the Initial Measurement Data to the condition defined in "IMD Created Condition Value".

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCACALINT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ScalarCalcFromInterval

## Scalar Estimation

This rule uses historical data for the same measuring component to derive an estimated value for a scalar Initial Measurement. Depending upon a VEE Rule configuration parameter, the routine uses historical data from a month ago followed by a year ago, or vice-versa. If the data for the first historical period turns out to be usable, the second historical period will not be evaluated. Whether historical data qualifies for use in estimation is configured via the rule parameters in conjunction with the below algorithm parameters.

The rule rejects consumption from a historical period as unusable for estimation if too great a portion of the period is covered by final measurements that are not high-quality. The first pair of parameter values are used when evaluating the acceptable System-Estimated Percentage for historical periods as configured on the VEE Rule. Once an estimated value is calculated, the routine backs into a reading (which involves backing out multipliers and, if the measuring component is subtractive, adding the result to the prior reading).

An interim High/Low can also be configured on this rule.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCALAREST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ScalarEstimation

## Scalar Profile Estimation


Copyright © 2026, Oracle and/or its affiliates.

This VEE rule arrives at a scalar estimate by looking at final measurements for a profile measuring component covering the same date range as the current Initial Measurement. The profile measuring component to be used as a source of measurement data is defined via a profile factor on the rule. This rule is meant primarily for a configuration in which the profile measuring components are interval, although the profile could be scalar as well.

If a measurement already exists in the current Initial Measurement with a condition that lies within the ranges specified in the pairs of algorithm parameters for "system estimate" and "regular", the routine does not attempt to estimate.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCAPROEST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ScalarProfileEstimation

## Scalar Proration

This rule prorates the value of a scalar reading that has two valid scalar readings on either side as boundaries.

If the scalar MC does have a related interval MC and 'Minimum Related Measurement Condition' is defined the algorithm will take into consideration any intervals during the proration period that meet a minimum condition quality defined by the 'Minimum Related Measurement Condition' when performing the proration. This is to keep the prorated scalar estimates in sync with any existing interval measurements. To do this the algorithm will augment the total duration and total quantity being prorated by the related interval MC's qualifying measurements. For example, take the following scenario:

- Scalar proration is executing for a scalar IMD for 01/01 12:00 AM - 01/02 12:00 AM (24 hours)
- The subsequent measurement is for 150 kWh and has a measurement date/time of 01/04 12:00 AM (72 hours)
- The related interval MC has measurements on 01/03 12:00 AM to 01/04 12:00 AM totally 100 kWh (24 hours)

Therefore the result of the calculation for the scalar IMD from 01/01 12:00 AM to 01/02 12:00 AM would be 25 kWh. Since the interval measurements covered 24 of the 72 total hours of the 150 kWh that would leave 50 kWh to be split among the remaining 48 hours. Which leaves 25 kWh for 01/01 12:00 AM to 01/02 12:00 AM and 25 kWh for 01/02 12:00 AM to 01/03 12:00 AM.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SCLRPRORT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ScalarProration

### Subtractive Interval Adjustment Rule

This rule adjusts qualifying consecutive intervals for a subtractive interval initial measurement (IMD) based on the known consumption between a set of actual readings. The readings can either be leveraged from other intervals within the IMD or from the final measurements when there are no suitable readings within the IMD. The adjustment is done by applying an adjustment factor to each of the qualifying intervals.

The rule supports adjusting multiple sets of qualifying consecutive intervals by calculating a consumption adjustment target and adjustment factor specific to each set of consecutive qualifying intervals.

Intervals are considered to be qualified for adjustment based on condition. A range of conditions is configured on the VEE Rule, and the rule adjusts only those intervals with conditions that lie within the configured range.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SUBINADJV Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-SubIntAdjustmentVEE

## Decision-Making VEE Rules

- Exception Handler
- Execute VEE Group
- Successful Termination
- VEE Group Matrix (Factor)

### Exception Handler


Copyright © 2026, Oracle and/or its affiliates.

This rule allows you to terminate the VEE process based on a configurable set of exceptions being present for the IMD. This rule also allows a unique To Do Type to be generated based on a group of exceptions.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-EEH Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D1-VEERuleExceptionHandler

## Execute VEE Group

Many implementations need to execute a group of validations for any data being received. For example, you might want to perform device identifier validations, multiplier checks, and UOM checks on all measuring components. One inefficient way to meet this requirement would be to repeat these three rules in multiple VEE groups. However, this solution becomes hard to maintain if changes to the rules are required (or if new "global rules" are introduced) as each group would have to be updated. Instead of this, you can use the "Execute VEE Group" rule to execute a referenced VEE Group. Using the example above, you could create a group called "Rules for All MCs" that contains a device identifier validations rule, a multiplier check rule, and a UOM check rule, and then reference the "Rules for All MCs" group in a Execute VEE Group rule.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-RFG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D1-VEERuleReferredVEEGroup

## Successful Termination

This rule allows VEE to be successfully terminated based on a list of exceptions.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-EST Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D1-VEERuleSuccessTermination

### VEE Group Matrix (Factor)

Another situation likely to occur in many implementations is where specific rules may need to be applied to measurement data based on specific criteria, such as geography. For example, some geographic territories may have unique VEE Rules in addition to rules that are applied to all geographic territories. The "VEE Group Matrix (Factor)" rule allows for a Factor to determine which VEE Rule gets executed based on defined characteristics.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AVEER-FCT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D1-VEERuleGroupFactor

**Prerequisite Setup**

Since the prerequisite setup items for this rule are more involved, the following procedure describes these in detail:

1. Create the Characteristic Type and Values to be used by the factor that will be referenced by the rule.
2. Create the Characteristic Source Algorithm to be used by the factor that will be referenced by the rule.
3. Create the VEE Groups to be associated to the characteristic values.
4. Create the Factor that will be referenced by the rule.
5. Create the Factor Values for the factor, each referencing an effective-dated characteristic value/VEE group pairings.
6. Create the rule, referencing the factor

# Measuring Component Statistics


Copyright © 2026, Oracle and/or its affiliates.

- Understanding Measuring Component Statistics
- Configuring Measuring Component Statistics

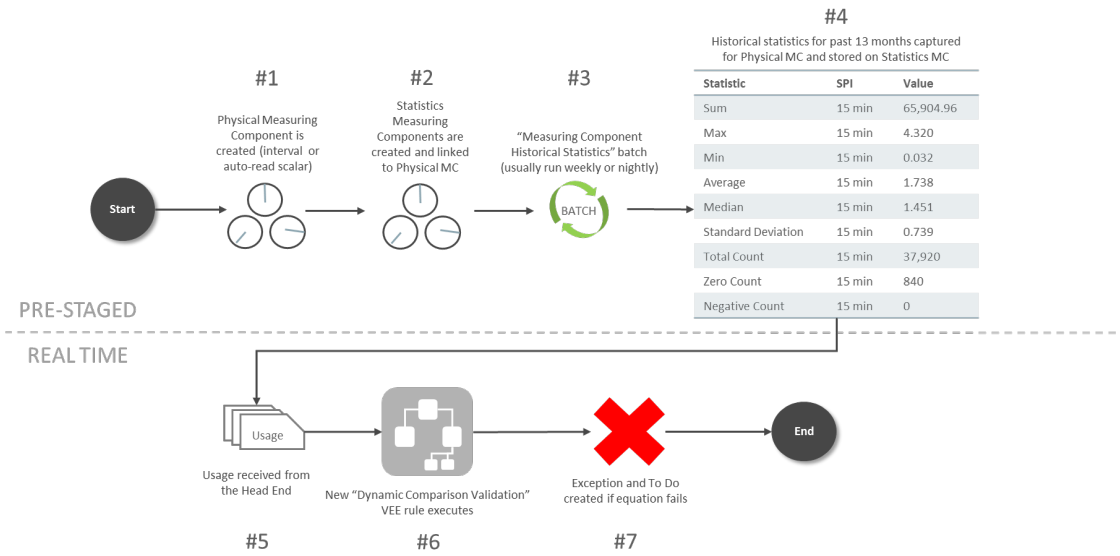
# Understanding Measuring Component Statistics

Oracle Utilities Meter Data Management provides a way to stage statistics for Measuring Components in order to speed up VEE processing. These statistics are primarily meant for use in conjunction with the Dynamic Comparison Validation VEE Rule, but could also be generated for other analytics or reporting.

Some of the potential benefits of using Measuring Component Statistics are:

- Reduce the data volume for VEE queries by up to 1000 times (this is most beneficial when replacing VEE Rules that query historical data such as High/Low)
- Offload heavier processing of historical data to "system downtimes" - potentially nights or weekends

Below is an overview example of how Measuring Component Statistics can be implemented along with the Dynamic Comparison Validation:




The steps for pre-staging Measuring Component Statistics are detailed below:

1. A Physical Measuring Component is created. The algorithm to trigger the creation of Statistics Measuring Components is only set on Interval or Auto-Read Scalar BOs as part of base product.
2. New Statistics Measuring Component are created and linked to the Physical Measuring Component.
3. The Measuring Component Historical Statistics Batch (D1-MCHS) is run based on the schedule you've defined. The suggestion is to run this either nightly or on the weekends during quieter times for overall system processing.
4. A measurement is created for the Statistics Measuring Component that holds historical statistics related to the Physical Meter for a specific period of time (in this example it's set for 13 months).

The steps for how VEE can take advantage of Measuring Component Statistics are detailed below:


Copyright © 2026, Oracle and/or its affiliates.

1. IMDs are received from the Head End.
2. The Dynamic Comparison Validation VEE Rule is configured to execute for the meter. This validation uses the historical statistics from the Statistics Measuring Component in order to perform its validation logic.
3. An exception and potentially a To Do is generated based on the failure of the VEE rule.

### Storage Impact from Statistics

Much consideration was given to the storage implications when designing Measuring Component Statistics. Below is an outline of the storage impact to customers who choose to calculate and store Channel Statistics. An example impact to a utility with 1 million Physical Meters (each with 4 channels of information) is shown in the far right column:

| Area | Additional Data in MDM | Example Impact (assuming utility has 1 million Physical Meters) |
|---|---|---|
| **Additional MCs**<br>*Assumption: Statistics are only configured on the primary billing channel MC Type for the majority of your Physical Meters.* | 1-3 extra MCs per Physical Meter<br>*Assumption: customers will only configure Statistics on their primary billing channel MC Type for the majority of their Physical Meters.* | 1 - 3 million extra MCs<br>*This increases the Measuring Component table by an extra 25% to 75% from its size without statistics.* |
| **Additional Measurements**<br>*If 'Statistics Retention Mode' configured as "Only Retain Most Recent Record".* | 1 extra Measurement per Physical Meter | 1 - 3 million extra Measurements for the life of MDM<br>*This data is a minuscule fraction compared to the overall AMI data footprint.* |
| **Additional Measurements**<br>*If 'Statistics Retention Mode' configured as "Retain Historical Records".* | 1 extra Measurement per Physical Meter per week | 1 - 3 million extra Measurements per week (52-156 million per year)<br>*This data is 0.2-0.5% of the size of the AMI data collected in that same week.* |

# Configuring Measuring Component Statistics

### Measuring Component Types

In order to start generating Measuring Component Statistics, two different configurations must occur for Measuring Component Types:

- New Measuring Component Types must be configured that leverage the **Channel Statistics Type** business object. These should be setup as the periods for which Measuring Component Statistics should be calculated. For example, if you would like to be able to reference both last month, last year, as well as the past 13 months as statistics periods, then 3 different Channel Statistics Types must be setup for this. These could be setup as follows:
- – "Last Month" Channel Statistics Type:
  - – Calculation Lag Days = 0
  - – Calculation Horizon Days = 30


Copyright © 2026, Oracle and/or its affiliates.

- "Same Month Last Year" Channel Statistics Type:
- Calculation Lag Days = 365
- Calculation Horizon Days = 395
- "Prior 13 Months" Channel Statistics Type:
- Calculation Lag Days = 0
- Calculation Horizon Days = 395

***NOTE:*** *it's recommended to set the Statistics Retention Mode to "Only Retain Most Recent Record" on the Channel Statistics Type to reduce the amount of data being stored in Oracle Utilities Meter Data Management.*

- For a Physical Measuring Component Type leveraging either "Interval Channel Type" or "Register - Auto-Read Type", the **Related Statistics Measuring Component Types** section must be filled in with the appropriate types that should be created for the Physical Measuring Component.

#### VEE Rule

If you have a desire to leverage Measuring Component Statistics for VEE, the Dynamic Comparison Validation VEE Rule section should be referenced. A number of detailed examples of how to implement this rule are provided in this section as well.

#### Related Batch Controls

There are a few batches directly involved with Measuring Component Statistics:

- **Measuring Component Historical Statistics** (D1-MCHS): this batch monitors any active Statistics Measuring Components to execute the logic for calculating a new set of historical statistics.
- **Statistics Measuring Component Creation** (D1-STMCC): this batch is used to create Statistics Measuring Components on any Physical Measuring Components that were created prior to configuring this module. This is especially useful for existing customers that want to implement Statistics Measuring Components for all of their devices.

> **Note**
>
> Refer to Understanding Measuring Component Statistics for more information.

# Defining Usage Options

- Usage Subscription Types
- Integrating Usage Transactions
- Usage Transaction Exception Types
- Usage Calculation Groups
- Usage Calculation Rules
- Usage Calculation Types
- Usage Subscription Quantity Types


Copyright © 2026, Oracle and/or its affiliates.

- Dynamic Option Types
- Contact Types
- Bill Cycle

# Usage Subscription Types

- Understanding Usage Subscription Types
- Configuring Usage Subscription Types

## Understanding Usage Subscription Types

Usage Subscription Types define a collection of properties defining a class of Usage Subscriptions. Usage Subscription types also control valid values for various attributes of Usage Subscriptions.

For a deeper functional understanding, refer to the About Usage Subscriptions or About Usage Calculation sections of the *Oracle Utilities Meter Solution Business User Guide*.

## Configuring Usage Subscription Types

This portal is used to display and maintain a Usage Subscription Type.

Refer to Understanding Usage Subscription Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Subscription Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Type List**: displays all of the Usage Subscription Types so you can choose the one you want to display in more detail
- **Usage Subscription Type**: shows the specific configuration for the selected Usage Subscription Type

The Usage Subscription Type defines a number of lists for "valid" objects that can be used in conjunction with the overall usage calculation process:

- Valid Usage Recipients
- Valid Service Point Types
- Valid Usage Calculation Groups
- Valid Dynamic Option Types

# Integrating Usage Transactions

- Requesting Usage Transactions from External Systems
- Processing and Sending Usage Transactions to External Systems
- Generating Usage Transactions in Oracle Utilities Meter Data Management


Copyright © 2026, Oracle and/or its affiliates.

# Requesting Usage Transactions from External Systems

When Oracle Utilities Meter Data Management is deployed with Oracle Utilities Customer Care and Billing usage transactions can be created via a request from Oracle Utilities Customer Care and Billing.

To invoke a usage transaction request, Oracle Utilities Customer Care and Billing invokes a Usage Transaction Request Inbound service script which invokes the Usage Transaction Seeder (D2-UsgTranSeeder) business object. This business object does the following:

- Determines the usage subscription ID based on an external usage subscription ID
- Determines the appropriate usage transaction business object to create. This uses the "How To Create Usage Subscription Related Information" processing method defined for the "Usage Transaction Creation" processing role on the service provider that represents the external system.

For a deeper functional understanding of Usage Transactions, refer to the About Usage Transactions section of the *Oracle Utilities Meter Solution Business User Guide*.

# Processing and Sending Usage Transactions to External Systems

When Oracle Utilities Meter Data Management is integrated with a customer information system (such as Oracle Utilities Customer Care and Billing), usage transaction information, including bill determinants and other data can be sent to the customer information system (again, such as Oracle Utilities Customer Care and Billing).

Usage transactions are sent to subscribing systems when they enter the "Sent" state. The "Send Usage" (D2-SEND-USG) Enter algorithm on the Sent state of the base package usage transaction business object (D2-UsageTransaction) determines the method used to send the information, based on the Usage Recipient (service provider) defined on the usage transaction's Usage Subscription. Usage transactions can be sent to service providers via either online real-time processing or periodically via batch processing.

### Online Real-Time Processing

To set up the service provider to support online real-time notification of usage transactions, do the following:

- Create one or more Outbound Message Types that reference the outbound message business object to be used to send usage transaction information to the external system. The base package include the following business object for Usage Transaction Outbound Message (D2-UsageTranOutboundMesg).
- Define a Message Sender that will be used to send the message to the external system.
- Add the outbound message type to the service provider's External System and reference the Message sender created above.
- Add a processing method to the service provider as follows:
  - **Processing Role**: Usage Transaction Notification - Online
  - **Processing Method**: How To Send US Related Information
  - **Status**: Active
  - **Default Processing Method**:
  - **Outbound Message Type**: the outbound message type created above


Copyright © 2026, Oracle and/or its affiliates.

- **Override Processing Method**: outbound message types for specific usage subscription types if applicable

For a deeper understanding of integrating usage transactions with a customer information or Oracle Utilities Customer Care and Billing, refer to the Configuring the Bill Determinant Interface in the Interface section.

### Batch Processing

If Oracle Utilities Meter Data Management is integrated with a customer information system (such as Oracle Utilities Customer Care and Billing) that requests bill determinants, usage transaction processing should be coordinated with the requesting system's billing process.

Requests from a billing system that have been indicated as "batch requests" (such as those produced by Oracle Utilities Customer Care and Billing batch billing process) accumulate in a "calculation deferred" state to be processed specially by the **Usage Transaction Calculate Defer Monitor** batch control (D2-UTCD).

To set up the service provider to support periodic batch processing of usage transactions, use a periodic monitor batch control. These batch programs should invoke the business objects that will contain the usage transaction information. The base package includes the following business object for this: Usage Transaction Outbound Message (D2-UsageTranOutboundMesg).

Next, setup the following configuration:

- Add a processing method to the service provider as follows:
  - **Processing Role**: Usage Transaction Notification - Batch
  - **Processing Method**: How To Send US Related Information
  - **Status**: Active
  - **Default Processing Method**:
  - **Batch Control**: the batch control created above
  - **Override Processing Method**: batch controls for specific usage subscription types if applicable

Errors encountered when processing Usage Transactions can be reprocessed with the following batches:

- Reprocessing usage transaction "seeders" in error (D2-UTSED)
- Reprocessing usage transactions in error (D2-UTID)

If unexpected errors occur that leave usage transactions in an unmonitored state, the **Usage Transaction Monitor** batch control (D2-UT), or one based on this batch control with parameter values tailored to any specific requirements, can be used to process those usage transactions.

For a deeper functional understanding of Usage Transactions, refer to the About Usage Transactions section of the *Oracle Utilities Meter Solution Business User Guide*.

## Generating Usage Transactions in Oracle Utilities Meter Data Management

There may be some cases where Usage Transactions should be generated in Oracle Utilities Meter Data Management rather than requested from an external system. Generation of these usage transactions are scheduled around the time when the meter installed on the service point is read.


Copyright © 2026, Oracle and/or its affiliates.

Base product delivered a batch program that reads all pending measurement cycle schedule routes with selection date on or before the business date and creates as SP measurement cycle schedule route business object instance that stages the creation of these usage transactions.

For additional details refer to the D1-CSPSR batch control and D1-SPMsrmtCycScheduleRoute business object.

# Usage Transaction Exception Types

- Understanding Usage Transaction Exception Types
- Configuring Usage Transaction Exception Types

## Understanding Usage Transaction Exception Types

Usage Transaction Exception Types define the groupings of exceptions for a Usage Transaction based on their functional similarity. This provides a way to define Usage Transaction Exceptions in a distinct enough way to understand the root issue that was generated from the usage calculation rule.

For a deeper functional understanding of Usage Calculation, refer to the About Usage Calculation section of the *Oracle Utilities Meter Solution Business User Guide*.

## Configuring Usage Transaction Exception Types

You ue the **Usage Transaction Exception Type** portal to display and maintain usage transaction exception types .

Refer to Understanding Usage Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Transaction Exception Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Transaction Exception Type List**: displays all of the Usage Transaction Exception Types so you can choose the one you want to display in more detail
- **Usage Transaction Exception Type**: shows the specific configuration for the selected Usage Transaction Exception Type

# Usage Calculation Groups

- Understanding Usage Calculation Groups
- Configuring Usage Calculation Groups

## Understanding Usage Calculation Groups

Usage calculation groups are collections of usage calculation rules that are used to either calculate bill determinants or validate bill determinants. During the usage transaction process, the system executes the usage calculation rules defined in the usage calculation group


Copyright © 2026, Oracle and/or its affiliates.

referenced on the usage subscription. The rules within a usage calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Usage calculation groups are associated with specific usage subscriptions and usage subscriptions types (or both). When assigned to usage subscriptions, usage calculation groups contain the usage calculation rules to be used to calculate usage / bill determinants. Usage calculation groups associated with usage subscription types are those groups considered valid for usage subscriptions of that type.

Usage calculation groups can also specify a list of device donfiguration types that are considered valid. Usage calculation groups should only be associated with usage subscriptions for service points related to device configurations of a valid device configuration type.

Usage calculation groups can also be referenced by the Execute Usage Calculation Group usage calculation rule.

For a deeper functional understanding of usage calculation, refer to the About Usage Calculation section of the *Oracle Utilities Meter Solution Business User Guide*.

## Configuring Usage Calculation Groups

This portal is used to display and maintain a usage calculation group.

Refer to Understanding Usage Calculation Groups for more information.

You can access the portal by selecting**Admin**, then **Usage**, then **Usage Calculation Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Calculation Group**: Defines basic information about usage calculation group
- **Usage Calculation Rules List**: lists the usage calculation rules belonging to the group
- **Referencing Usage Calculation Rules List**: lists the usage calculation rules that reference the group
- **Referencing Usage Subscription Type List**: lists the usage subscription types that reference the group
- **Referencing Usage Subscriptions List**: lists the usage subscriptions that reference the group

# Usage Calculation Rules

- Understanding Usage Calculation Rules
- Configuring Usage Calculation Rules
- Detailed Configuration Examples

## Understanding Usage Calculation Rules

Usage calculation rules are standard and custom rules that perform calculations on measurement data to generate bill determinants and other values used by external systems, such as billing systems, customer information systems, etc. Usage calculation rules are created for a specific usage calculation group. For example, if you were configuring two usage


Copyright © 2026, Oracle and/or its affiliates.

calculation groups and both included a specific usage calculation rule, you would need to create two instances of the usage calculation rule, one for each group.

The base package includes rules that calculate common bill determinants including:

- Scalar reads
- Time-of-use consumption (by applying a time-of-use map to an interval channel)
- Interval curves (either real or derived)
- Virtually anything else that can be calculated from the information in the system

For a deeper functional understanding of usage calculation, refer to the About Usage Calculation section of the *Oracle Utilities Meter Solution Business User Guide*.

### Usage Transaction Exceptions

On almost every usage calculation rule, the failure of the rule results in a usage transaction exception and the Usage Transaction Exception Type for the failure can be configured on the rule. These usage transaction exception types can also be set to a specific Exception Severity:

- **Information**: Used to highlight minor issues, but not sufficient to cause the usage transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues
- **Issues**: Used to report a problem that will prevent the usage transaction from being sent. Multiple "issue exceptions" can be created during usage transaction processing. If at least one issue exists after all rules have been applied, the usage transaction is transitioned to a failure state requiring review and approval.
- **Terminate**: Used to report a severe issue that will cause the usage calculation process to stop and the usage transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop on for a Usage Transaction). This should be used for cases where manual override / approval isn't accurate. For example, a "Curve Not Continuous" error that says the interval data doesn't cover the full usage period should always be set to Terminate as an Exception Severity.

### Rounding Usage Transaction Service Quantities

Several usage calculation rules provide a means for defining the manner in which service quantities are rounded during usage calculations. Service quantity rounding can be defined using the **SQ Rounding Details** which allow configuration of the method by which quantities are rounded (Up, Down, or Nearest) and the number of decimal positions retained after rounding.

### Creating Measurement Data Snapshots

Daily Scalar Calculation Rule, Get Interval, Get Item Counts and Consumption, and Get Scalar Details usage calculation rules can be configured to create measurement data snapshots during usage transaction processing. Measurement data snapshots are used by data aggregation in Market Settlements Management.

Measurement data snapshot configuration is done in the **Extract To Measurement Data Snapshots** section, and includes the following options:

- **Measurement Data Snapshot Type**: The measurement data snapshot type for measurement data snapshots created by the usage rule.
- **Usage Transaction ID To Use**: A flag that designates the type of ID to use to identify the usage transaction on the measurement data extract. Options include:


Copyright © 2026, Oracle and/or its affiliates.

  - **Usage Transaction ID**: The usage transaction's ID
  - **Usage Transaction External ID**: The external ID for the usage transaction. Used when the usage transaction is triggered by external system such as a CIS (or by the billing functionality in Customer To Meter)

- **Extract To Different Usage Subscription**: A flag that designates whether or not the measurement data snapshot should be extracted to a different usage subscription, such as one used by Market Settlements Management. Options include:
  - **No**: The extract will reference the usage subscription for which usage calculations are being processed. This option is used when Meter Data Management and Market Settlements Management are installed in different application environments. In this case, the extract can be exported from Meter Data Management for subsequence import into Market Settlements Management.
  - **Yes**: The extract will reference a different usage subscription such as one used by Market Settlements Management. This option is used when Meter Data Management and Market Settlements Management are installed in the same application environment. In this case, the measurement data snapshot is "exported" within the system to the Settlement Item, where it can be used in data aggregation processing. When this option is selected, the **Source Identifier Type** and **Target Identifier Type** fields must be configured to define how the system will determine the correct settlement usage subscription to use.
- **Irregular Data Flag Mapping**: Ranges of condition codes used to flag data as "irregular". The value of the **Irregular Data Flag** (based on the IRREGULAR_DATA_FLG lookup) is used to flag data that falls with each range configured. A blank **Irregular Flag** value denotes that the condition range is considered to be "Regular".

See About Measurement Data Snapshots for more information about measurement data snapshot creation and processing.

#### Usage Calculation Rules and Anchor Time

Some usage calculation rules have been enhanced to support usage calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing measurement data and usage transactions. When processing usage calculations for measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the measuring component type.

## Configuring Usage Calculation Rules

This portal is used to display and maintain a usage calculation rule.

Refer to Understanding Usage Calculation Rules, Understanding Usage Calculation Groups, and Understanding Usage Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Calculation Rule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Calculation Rule**: Defines the usage calculation rule, including parameters used when executing the rule
  - Click **Show All Rules** to open the **VEE Rules** List zone, which lists all of the usage calculation rules in the usage calculation group to which the current rule belongs.


Copyright © 2026, Oracle and/or its affiliates.

- **Eligibility Criteria List**: Lists the eligibility criteria defined for the rule

### Pre-Calculation Usage Calculation Rules

The following is a list of the pre-calculation usage calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Alignment and Delay Usage Calculation Rule | Interval or Scalar | This rule can be used to handle two main needs:<br>1) aligns all measuring components for a Usage Subscription to the same date (whether on the same device or separate devices)<br>2) delays the usage transaction until the end of the retry window based on the quality of available data and other configured parameters. |
| Check Existence of Installed Device | Interval or Scalar | This rule checks for the existence of a device installed on the Usage Subscription's Service Point for the usage period. In the case of multi-items this rule also checks that they are effective for the usage period. |

### Calculation Usage Calculation Rules

The following is a list of the calculation usage calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Apply Math (Interval Data) | Interval | This rule is used to perform calculations on interval data and stores the results in the usage transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions).<br>This rule provides aggregated usage for all selected interval measuring components (filter by TOU, SQI & UOM) associated to a usage subscription.<br>This rule can also multiply total usage by a factor using a custom formula. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Coincident Peak and Individual SP Demand | Interval | This rule is used to calculate the coincident peak demand for all of the service points on the usage subscription and the demand values for each service point at the coincident peak. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point). |
| Daily Scalar Usage Calculation Rule | Scalar | This rule is used to calculate usage of daily scalar measuring components installed in the Service Points associated with a Usage Subscription for the specified usage period. It creates bill determinants by taking the difference between the beginning and ending reading for the bill period.<br>This rule can also be used to provide register readings by measuring component.<br>For consumption values, only the beginning and ending readings are exported<br>This rule supports date breaks (the normal Get Scalar Data rule does not). |
| Get Interval Data | Interval | This rule is used to get interval quantities from interval measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Interval' usage period.<br>This rule retrieves the interval data for measuring components associated to a usage subscription by TOU, SQI and UOM.<br>This rule converts the interval data to another interval length or unit of measure. |
| Get Item Counts and Consumption | Interval or Scalar | This rule finds item-based and multi-item-based Service Points linked to the Usage Subscription for the current usage transaction, summarizes the item counts by item type and Service Point, and calculates item-based consumption. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Get Scalar Details | Scalar | This rule is used to get usage from scalar measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Scalar' usage period.<br>This rule creates bill determinants by summing all scalar readings for the bill period. This rule can also be used to provide register readings by measuring component. All readings are exported by this rule.<br>Note: This rule is used for traditional monthly read meters. |
| Get Subtractive Interval Details | Interval | This rule is used to get interval quantities from subtractive interval measuring components installed on the service points linked to the usage subscription for the specified 'interval' usage period.<br>It also identifies the start and stop readings for each usage period using subtractive interval readings. |
| Get TOU Mapped Usage | Interval | This rule is used to get time of use quantities from interval measuring components for devices installed at the Service Points linked to the Usage Subscription for the specified 'Interval' usage period. |
| Interval Tier Calculation | Interval | This rule calculates the difference between a source and reference vector.<br>This rule loops through each tier that is configured and calculates the imbalance amount associated to that tier level.<br>This rule breaks down that difference into one to many positive or negative tiers, and create a service quantity for each tier calculated. |
| Profile Accumulation | Interval | This rule is used to manipulate a customer's interval data by adding other vectors to it. Those other vectors are derived from a list of profile factors and corresponding characteristic values stored in a list on the usage transaction. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Rolling Demand | Interval | This rule is used to calculate rolling demand for all of the service points on the usage subscription. Rolling demand can be based on a rolling consumption total, rolling consumption average, or coincident peak demand. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point). |
| Round and Adjust Usage | Interval or Scalar | This rule copies identified source and target Service Quantities and inserts these as Service Quantities that are rounded and adjusted. |
| Vector and Service Quantity Math | Interval | This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations.<br>Typical uses:<br>Perform math using interval data, e.g., take the difference between two curves, find max values, find coincident peaks, multiply a curve by a value, apply TOU maps, etc.<br>Define complex formulas using various interval curves, profile factor values or calculated service quantities (bill determinant values).<br>Support math functions: sin, cos, square root, etc.<br>Store derived curves in memory that can be used in subsequent calculations<br>Please note, this rule is not as efficient as other rules. |

### Post-Calculation Usage Calculation Rules

Below is a list of the post-calculation usage calculation rules provided as part of base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Usage High/Low Rule | Interval or Scalar | This rule compares the Service Quantities of the Usage Transaction to historical values. If the current value is too high or too low compared to historical data then an exception is thrown. |
| Validate Against Tolerance | Interval or Scalar | This rule is used to validate the calculated usage against a tolerance value. The tolerance value may either come from the specified value or tolerance factor defined in the usage calculation rule. |
| Business Flag Hold | Interval or Scalar | This rule can stop a usage transaction from proceeding when there have been business flags for the applicable service points. The hold can either be indefinite or set to expire a configurable amount of time prior to the calculation window ending. |

### Decision-Making Usage Calculation Rules

There are usage calculation rules delivered as part of the base product that help with decision-making when executing running the usage calculation process. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Usage Calculation Rule Name | Purpose |
|---|---|
| Execute Usage Calculation Group | This rule performs a call to execute a separate usage calculation group which includes execution of all usage calculation rules within that group. |
| Exception Handler | This rule is used to terminate execution of the usage processor if exception count criteria specified in the rule is met. |

### Advanced Aside: Using Factors For Variables

A situation common in some implementations involves converting one unit of measure (UOM) to another. However, the conversion factor used in conversions of this can differ based on many different types of criteria, such as the location of the service point or other characteristics. This sort of calculation can be implemented as a usage calculation rule that accumulates consumption for one UOM and converts the consumption to a different UOM by applying a factor to it.

Factors used for this purpose have a Factor Class of "Number," and use some unique rules:

- Number factors reference a characteristic type (with pre-defined values).
- Number factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.

Factor values for a Number factor are effective-dated pairings of a characteristic value and a corresponding value. Because these pairings are effective-dated, the value returned from the factor can change over time for each characteristic value. At run time, the rule retrieves /


Copyright © 2026, Oracle and/or its affiliates.

derives the characteristic value for the factor's characteristic type and then finds the value associated with the respective characteristic value. Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute**: you could create a rule that retrieves a specific value based on the location of a service point.
- **Dynamic Attribute**: you could create a rule that retrieves a percentage value based on the amount the customer conserved as compared to the same period in the prior year, returning one value if the amount conserved is between 5% and 10%, another value if the amount conserved is between 10% and 20%, and yet a third value if the amount conserved is greater than 20%. The amount conserved is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and returns the appropriate value. In this example, if the amount conserved was anything less than 5%, no percentage value would be returned.
- Pre-Calculation Usage Calculation Rules
- Calculation Usage Calculation Rules
- Post-Calculation Usage Calculation Rules
- Decision-Making Usage Calculation Rules

## Pre-Calculation Usage Calculation Rules

- Alignment and Delay Usage Calculation Rule
- Check Existence of Installed Device

### Alignment and Delay Usage Calculation Rule

This rule attempts to delay usage calculation until high quality data for calculating bill determinants have become available. It is especially useful for usage subscriptions having multiple sources of usages such as multiple service points or multiple measuring components on the installed meter. The rule ensures that the usage calculated is based on a common date or alignment date, identified for these sources.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ALGNDELAY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-AlignmentandDelay

### Check Existence of Installed Device

This rule checks for the existence of a device installed on the Usage Subscription's Service Point for the usage period. In the case of multi-items this rule also checks that they are effective for the usage period.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-CHKEXTDVC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ChkExistenceofInstalledDvc

## Calculation Usage Calculation Rules

- Apply Math (Interval Data)
- Coincident Peak and Individual SP Demand
- Daily Scalar Usage Calculation Rule
- Get Interval Data
- Get Item Counts and Consumption
- Get Scalar Details
- Get Subtractive Interval Details
- Get TOU Mapped Usage
- Interval Tier Calculation
- Profile Accumulation
- Profile Service Quantity
- Rolling Demand
- Round and Adjust Usage
- Vector and Service Quantity Math

### Apply Math (Interval Data)

This rule is used to perform calculations on interval data and it stores the results in the usage transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions).

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-APPMATHIN Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.


Copyright © 2026, Oracle and/or its affiliates.

**Business Object**

D2-ApplyMathInt

**Example Scenarios**

Below are some example scenarios that can be achieved based on configuration of this rule.

**Scenario 1:** Get the total KWH for the period

- **Calculation Type** = Single Variable
- **Variable Name** = V1, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KWH
  - Set Function = Sum

**Scenario 2:** Get the higher value between the total KWH or total KVARH

- **Calculation Type** = Math Function
- **Math Function** = Max
- **Variable Name** = V1, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KWH
  - Set Function = Sum
- **Variable Name** = V2, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KVARH
  - Set Function = Sum

**Scenario 3:** Total KWH multiplied with a factor value

- **Calculation Type** = Mathematical Expression
- **Mathematical Expression** = V1 * V2
- **Variable Name** = V1, where the following is configured:
  - Variable Type = Channel Accumulation
  - UOM = KWH
  - Set Function = Sum
- **Variable Name** = V2, where the following is configured:
  - Variable Type = Factor
  - Factor = factor code that holds the multiplier

## Coincident Peak and Individual SP Demand

This rule is used to calculate the coincident peak demand for all of the service points on a usage subscription and the demand values for each service point at the coincident peak. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point).


Copyright © 2026, Oracle and/or its affiliates.

### Interval Data Retrieval Parameters

This rule can be configured to use specific data either directly from a specific device and measuring component on the service point or calculated from more than one devices and/or measuring components on the service point. Each type of data is referred to as a "channel" of data. For example, a service point with solar panels might have separate channels for "Consumed" and "Generated" usage, and the rule could be configured to use data from either or based on calculations involving both.

Parameters used to retrieve interval data for this rule include:

- **Check Missing Intervals**: A flag that indicates if the rule should check for missing intervals during the calculation period.
- **No Meter Condition**: The measurement condition assigned to intervals inserted to fill gaps during periods in which when no meter is installed. If not set, the measurement condition defaults to "No Read - System" (100000).
- **SP Device Filter**: An algorithm used to filter the service point device configuration that will be processed by the rule.
- **Common UOM**: The unit of measure that is common among the multiple channels of data used by the rule, if applicable
- **Common Interval Size**: The interval size used to align service point data of different interval sizes. For instance, if a service point has channels measuring data in both 1-hour and 15-minute intervals, setting this to 01:00:00 (1 hour) would scale the 15-minute data to hourly intervals.
- **Channels**: A list of the data channels measured at the service point. Each should be assigned a variable name and the appropriate UOM, TOU, and SQI.
- **Aggregation Method**: The method by which data from multiple channels will be aggregated. Channels can be combined using a simple formula or using a conditional formula.

To define a single channel for the rule to use, configure the "Vector 1 (IV1)" variable in the **Channels** section and reference that variable in a simple formula.

To configure the rule to use multiple channels, define each as a variable in the **Channels** section and configure either a simple or conditional formula to perform any needed evaluations or calculations on the channels.

### Defining Formula Operands

Operands used in both simple and conditional formulas are defined in the Operand Lookup (D1-CoincidentPeakOperandLookup) extendable lookup. Operands used to represent channels should use the values in parentheses in the **Variables** drop-down list in the **Channels** section. For instance, the operand for "Vector 1 (IV1)" should be "IV1", the operand for "Vector 2 (IV2)" should be "IV2", and so on.

Operands can also be simple formulas based on the variables defined in the **Variables** drop-down list in the **Channels** section. For instance, a simple formula in which the values from Vector 1 are subtracted from the values in Vector2 could be defined as "IV2 - IV1".

### Calculation Modes

This rule has two main modes: calculating demand for individual service points on a usage subscription (Individual SP Demand), or calculating the coincident peak among all of the service points on a usage subscription (Coincident Peak). The coincident peak calculation can be performed on either all readings, or those designated as "on-peak" or "off-peak" via time of


Copyright © 2026, Oracle and/or its affiliates.

use mapping using the Coincident On-Peak or Coincident Off-Peak function. See Defining Time of Use Mapping Options for more information about time of use mapping.

**Individual SP Demand**

The "Individual SP Demand" **Calculate Function** identifies the peak demand for each service point on a usage subscription.

For example, the table below lists sample interval data for three service points belonging to single usage subscription for a 12 hour calculation period. This rule would identify the highest value (peak) for each service point within the calculation period (indicated in bold).

| Date/Time | Service Point 1 | Service Point 2 | Service Point 3 |
|---|---|---|---|
| 10/27/2022 12:00 PM | 12 | 11 | 13 |
| 10/27/2022 1:00 PM | 10 | 12 | 12 |
| 10/27/2022 2:00 PM | 11 | 14 | 14 |
| 10/27/2022 3:00 PM | 13 | 13 | 14 |
| 10/27/2022 4:00 PM | **14** | 14 | 14 |
| 10/27/2022 5:00 PM | 13 | **15** | 12 |
| 10/27/2022 6:00 PM | 12 | 13 | **15** |
| 10/27/2022 7:00 PM | 11 | 12 | 13 |
| 10/27/2022 8:00 PM | 12 | 13 | 12 |
| 10/27/2022 9:00 PM | 13 | 14 | 12 |
| 10/27/2022 10:00 PM | 12 | 14 | 13 |
| 10/27/2022 11:00 PM | 12 | 12 | 12 |

This rule would return the following values in the usage transaction service quantities list, which represent the peak values for each service point during the calculation period:

| Date/Time | Demand |
|---|---|
| 10/27/2022 4:00 PM | 14 |
| 10/27/2022 5:00 PM | 15 |
| 10/27/2022 6:00 PM | 15 |

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

**Coincident Peak / Coincident On-Peak / Coincident Off-Peak**

The "Coincident Peak" (or “Coincident On-Peak” or “Coincident Off-Peak") **Calculate Function** identifies the coincident peak demand for all of the service points on the usage subscription and the demand values for each service point at the coincident peak. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point).

The data used in the calculation is based on the specific **Calculate Function** selected:

- **Coincident Peak**: Evaluates all measurements matching the configured unit of measure


Copyright © 2026, Oracle and/or its affiliates.

- **Coincident On-Peak**: Evaluates measurements matching the configured unit of measure that falls with the "on-peak" TOU period defined in the **Results** section based on a specified TOU Map Template.
- **Coincident Off-Peak**: Evaluates measurements matching the configured unit of measure that falls with the "off-peak" TOU period defined in the **Results** section based on a specified TOU Map Template.

For example, the table below lists sample interval data for three service points belonging to single usage subscription for a 12 hour calculation period. This rule would identify the highest total for all of the service points, as well as the individual values for each service point at the time of the coincident peak within the calculation period (indicated in bold).

| Date/Time | Service Point 1 | Service Point 2 | Service Point 3 | Total |
|---|---|---|---|---|
| 10/27/2022 12:00 PM | 12 | 11 | 13 | 36 |
| 10/27/2022 1:00 PM | 10 | 12 | 12 | 34 |
| 10/27/2022 2:00 PM | 11 | 14 | 14 | 39 |
| 10/27/2022 3:00 PM | 13 | 13 | 14 | 40 |
| **10/27/2022 4:00 PM** | **14** | **14** | **14** | **42** |
| 10/27/2022 5:00 PM | 13 | 15 | 12 | 40 |
| 10/27/2022 6:00 PM | 12 | 13 | 15 | 40 |
| 10/27/2022 7:00 PM | 11 | 12 | 13 | 36 |
| 10/27/2022 8:00 PM | 12 | 13 | 12 | 37 |
| 10/27/2022 9:00 PM | 13 | 14 | 12 | 39 |
| 10/27/2022 10:00 PM | 12 | 14 | 13 | 39 |
| 10/27/2022 11:00 PM | 12 | 12 | 12 | 36 |

This rule would return the following values in the usage transaction service quantities list, which represent the total at the coincident peak and the peak values for each service point at the coincident peak during the calculation period:

| Date/Time | Coincident Peak Demand | SP 1 Demand | SP 2 Demand | SP 3 Demand |
|---|---|---|---|---|
| 10/27/2022 4:00 PM | 42 | 14 | 14 | 14 |

> **Note**
>
> Values for individual service points are for information and reporting purposes only, and cannot be used for billing purposes.

### Notes and Examples

### Notes and Recommendations

- When using the "Coincident On Peak" or "Coincident Off Peak" **Calculate Function**, you must specify a TOU map template in the **On Peak Off Peak Schedule** drop-down list in the **Calculation Details** section, and specify the appropriate time of use period (On-Peak or Off-Peak, respectively) in the **TOU** field in the **Results** section.
- A recommended practice is to configure a Service Quantity Identifier (SQI) to uniquely identify coincident peak data and specify it in the **SQI** field in the **Results** section.


Copyright © 2026, Oracle and/or its affiliates.

- Additional detail on the logic of this rule can be found in the Detailed Description of the D1-CALDEMAND Algorithm Type.

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D1-CoincidentPeakSPDemandRule

**Example Scenarios**

Below are some example scenarios that can be achieved based on configuration of this rule.

**Scenario 1**: Calculate demand for individual service points (similar to the example above, based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 01:00:00
  - **Channels**:
    - **Variable 1**:
      - **Variable Name**: Variable 1 (IV1)
      - **UOM**: Kilowatt-Hour
  - **Aggregation Method**:
    - **Formula Source**: Simple Vector Formula
    - **Simple Vector Formula**: IV1
- **Calculation Details**:
  - **Calculate Function**: Individual SP Demand
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**:
  - **SQI**:
  - **Common Interval Size**: 01:00:00

**Scenario 2**: Calculate demand for individual service points (based on two channels)

In this scenario, the service point has two channels, one for consumed energy, and one for generated energy (from a solar panel). If the consumed value is greater than the generated value, subtract the generated value from the consumed value. If the generated value is greater than the consumed value, use zero (0).

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 01:00:00


Copyright © 2026, Oracle and/or its affiliates.

- **Channels**:
  - **Variable 1**:
    - **Variable Name**: Variable 1 (IV1)
    - **UOM**: Kilowatt-Hour
    - **TOU**:
    - **SQI**: Consumed
  - **Variable 2**:
    - **Variable Name**: Variable 2 (IV2)
    - **UOM**: Kilowatt-Hour
    - **TOU**:
    - **SQI**: Generated
- **Aggregation Method**:
  - **Formula Source**: Conditional Vector Formula
  - **Conditional Vector Formula**:
    - **Sequence**: 10
    - **Left Operand**: IV1
    - **Criteria Operator**: >
    - **Right Operand**: IV2
    - **True Action**: Apply True Formula
    - **True Formula**: IV1-IV2
    - **False Action**: Apply False Formula
    - **False Formula**: 0

- **Calculation Details**:
  - **Calculate Function**: Individual SP Demand
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**:
  - **SQI**:

**Scenario 3**: Coincident Peak (similar to the example above, based a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 01:00:00
  - **Channels**:
    - **Variable 1**:
      - **Variable Name**: Variable 1 (IV1)


Copyright © 2026, Oracle and/or its affiliates.

        - **UOM**: Kilowatt-Hour
    - **Aggregation Method**:
        - **Formula Source**: Simple Vector Formula
        - **Simple Vector Formula**: IV1
- **Calculation Details**:
    - **Calculate Function**: Coincident Peak
- **Results**:
    - **UOM**: Kilowatt
    - **TOU**:
    - **SQI**: Coincident Peak

**Scenario 4**: Coincident On-Peak (based on a single channel)

- **Interval Data Retrieval Parameters**:
    - **Check Missing Intervals**: Yes
    - **No Meter Condition**: No Read - System
    - **Common UOM**: Kilowatt-Hour
    - **Common Interval Size**: 01:00:00
    - **Channels**:
        - **Variable 1**:
            - **Variable Name**: Variable 1 (IV1)
            - **UOM**: Kilowatt-Hour
    - **Aggregation Method**:
        - **Formula Source**: Simple Vector Formula
        - **Simple Vector Formula**: IV1
- **Calculation Details**:
    - **Calculate Function**: Coincident On Peak
    - **On Peak Off Peak Schedule**: <TOU map template>
- **Results**:
    - **UOM**: Kilowatt
    - **TOU**: On Peak
    - **SQI**: Coincident On Peak

## Daily Scalar Usage Calculation Rule

This rule is used to calculate usage of daily scalar measuring components installed in the Service Points associated with a Usage Subscription for the specified 'Interval' usage period. Only automatically read scalar measuring components with value identifier that matches usage calculation rule's UOM/TOU/SQI will be processed.

If configuring this rule with interval and scalar usage calculation rules under a usage calculation group, it should be executed after interval usage calculation rule and before the scalar usage calculation rule.


Copyright © 2026, Oracle and/or its affiliates.

This rule calculates usage based on the beginning and ending reading values -- ignoring intermediate readings. This rule will not estimate data based on the usage transaction estimate flag. Frequently read scalar devices must use periodic estimation processes.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETFRESCR Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-CalFrequentlyReadScalar

## Get Interval Data

This rule is used to get interval quantities from interval measuring components installed in the service points linked to the usage subscription for the specified 'Interval' usage period. Only measuring components that match the UOM/SQI defined in the usage calculation rule are processed. Measurements within the period are stored in the usage transaction's service quantities' interval data list. The SQ entry's quantity is calculated based on the **Calculate Function** defined in the usage calculation rule. This is done for every entry in the usage period list. Each service quantity recorded by this rule is linked to a service point and measuring component.

By supplying the Axis Conversion parameters on the usage calculation rule, this algorithm will convert the identified measuring component's consumption into the supplied UOM and interval size prior to storing the consumption into the service quantity's interval data list.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETINTDAT Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-GetIntervalData

## Get Item Counts and Consumption

This rule finds item-based and multi-item-based Service Points linked to the Usage Subscription for the current usage transaction, summarizes the item counts by item type and Service Point, and calculates item-based consumption. Once the item detail entries are created, the rule processes goes through them to create usage period SQs.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETITEMCC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-GetItemCountsConsumption

## Get Scalar Details

This rule is used to get usage from scalar measuring components installed in the Service Points linked to the Usage Subscription for the specified 'Scalar' usage period. By default all scalar measuring components are processed but if specific UOMs/TOUs/SQIs are defined in the usage calculation rule then only applicable measuring components are processed. Measurements within the period are retrieved. The usage transaction request may indicate whether or not 'Estimate' measurements are allowed. If the usage transaction request indicates that ‘Estimate’ measurements are allowed but an estimated end measurement cannot be found an Estimate initial measurement is created.

The measurement details are stored in the usage transaction's scalar details. The usage is also stored in the usage transaction's service quantities unless otherwise specified in the usage calculation rule (using Build Service Quantity indicator).

This rule can be configured to perform measurement quality assessment, which will result in the population of the measurement quality list with the quantities and corresponding conditions of those measurements (regular, estimated, etc.).

**Please note:** this rule should not be used with readings occurring daily or more frequently as it will retrieve all of the readings to store in the Usage Transaction. The Daily Scalar usage calculation rule may be a better option for this use case.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETSCALAR Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-GetScalar

## Get Subtractive Interval Details

This rule is used to calculate the usage for a subtractive interval measuring components. This differs from the Get Interval Data usage calculation rule in that it also retrieves a start and stop reading for each usage period it calculates. The start and stop readings are then used to


Copyright © 2026, Oracle and/or its affiliates.

provide entries into the usage scalar details table which are then made available in the usage transaction sent to the billing system for bill presentment.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-SUBINTADJ Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-GetSubtractiveInterval

## Get TOU Mapped Usage

This rule is used to get time of use quantities from interval measuring components for devices installed at the Service Points linked to the Usage Subscription for the specified 'Interval' usage period. Only measuring components that match the UOM/SQI defined in the usage calculation rule instance are processed.

Measurements within the period are mapped to time of use quantities based on the TOU map defined on the usage calculation rule. If dynamic options are specified in the referenced TOU map and if there are dynamic option events in effect within the usage period then the TOU map associated with the dynamic option is used for the entire dynamic option event period. This calculation is performed for every usage period requested.

The calculated time of use quantities are stored in the usage transaction's service quantities per Service Point and measuring component.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETTOUUSG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-GetTOUUsage

## Interval Tier Calculation

This rule calculates the difference between a source and reference vector. It then breaks that difference into one to many positive or negative tiers. For each tier calculated it will create a service quantity with the optional ability to create additional service quantities for the total of all positive tiers and the total of all negative tiers. This rule will loop through each tier that is configured and calculate the imbalance amount associated to that tier level.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-GETINTIER Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-IntervalTierCalculation

## Profile Accumulation

This rule is used to manipulate a customer's interval data by adding other vectors to it. Those other vectors are derived from a list of profile factors and corresponding characteristic values stored in a list on the usage transaction.

The typical application of this rule is for customer self-service rate compare, in which a self-service user has selected a set of usage adjustments to apply to their historical consumption to assess the effect on the amount consumed.

This rule algorithm retrieves the customer's usage (the source UOM) for the usage period using one of two options:

- A channel linked to a Usage Subscription
- A usage transaction service quantity

Then it takes each entry in the usage transaction's profile factor list, finds the profile measuring component for each, retrieves the measurement data for the profile, applies axis conversion to align it to the source UOM and target interval size, and adds it to the source vector. Each profile's data is added progressively to arrive at a final vector. The final vector may then be TOU-mapped, and the vector itself may be preserved or discarded.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-DYNPRFLAC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ProfileAccumulation

## Profile Service Quantity

This rule intervalizes scalar service quantity values based on a profile measuring component.

This rule first retrieves scalar service quantities, identified by the rule's UOM/TOU/SQI. If multiple service quantities exist for the service point they will be summarized at the service point level. In the case of a split or broken usage period for a service point, the usage period is


Copyright © 2026, Oracle and/or its affiliates.

merged and the quantities are summarized. Each service quantity is subject to intervalization. The profile measuring component is then derived using the profile factor supplied with either the usage subscription or service point as characteristic sources. If the profile is not available, an exception will be created using the exception details on the rule.

If a dynamic option type is provided, the events associated with it are retrieved for the usage transaction period. Any intervalized quantities that coincide with the dynamic option events start/end date time are then recorded in a new usage period service quantity of usage type “Correlated SQ”.

If no dynamic option type provided, the intervalized quantities are recorded in a new usage period service quantity of usage type “Profile SQ”.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-PRFSVCQTY Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ProfileServiceQuantity

## Rolling Demand

This rule is used to calculate rolling demand for all of the service points on the usage subscription. Demand is calculated from consumption. In electric service markets, these are typically represented by kilowatts (kW) and kilowatt hours (kWh) respectively, where demand values are calculated by multiplying consumption interval values by the number of intervals per hour. For example, the demand associated with a fifteen (15) minute interval value of 25kWh would be 100kW (25 x 4 intervals per hour).

**Interval Data Retrieval Parameters**

This rule can be configured to use specific data either directly from a specific device and measuring component on the service point or calculated from more than one devices and/or measuring components on the service point. Each type of data is referred to as a "channel" of data. For example, a service point with solar panels might have separate channels for "Consumed" and "Generated" usage, and the rule could be configured to use data from either or based on calculations involving both.

Parameters used to retrieve interval data for this rule include:

- **Check Missing Intervals**: A flag that indicates if the rule should check for missing intervals during the calculation period.
- **No Meter Condition**: The measurement condition assigned to intervals inserted to fill gaps during periods in which when no meter is installed. If not set, the measurement condition defaults to "No Read - System" (100000).
- **SP Device Filter**: An algorithm used to filter the service point device configuration that will be processed by the rule.
- **Common UOM**: The unit of measure that is common among the multiple channels of data used by the rule, if applicable


Copyright © 2026, Oracle and/or its affiliates.

- **Common Interval Size**: The interval size used to align service point data of different interval sizes. For instance, if a service point has channels measuring data in both 1-hour and 15-minute intervals, setting this to 01:00:00 (1 hour) would scale the 15-minute data to hourly intervals.
- **Channels**: A list of the data channels measured at the service point. Each should be assigned a variable name and the appropriate UOM, TOU, and SQI.
- **Aggregation Method**: The method by which data from multiple channels will be aggregated. Channels can be combined using a simple formula or using a conditional formula.

To define a single channel for the rule to use, configure the "Vector 1 (IV1)" variable in the **Channels** section and reference that variable in a simple formula.

To configure the rule to use multiple channels, define each as a variable in the **Channels** section and configure either a simple or conditional formula to perform any needed evaluations or calculations on the channels.

**Defining Formula Operands**

Operands used in both simple and conditional formulas are defined in the Operand Lookup (D1-CoincidentPeakOperandLookup) extendable lookup. Operands used to represent channels should use the values in parentheses in the **Variables** drop-down list in the **Channels** section. For instance, the operand for "Vector 1 (IV1)" should be "IV1", the operand for "Vector 2 (IV2)" should be "IV2", and so on.

Operands can also be simple formulas based on the variables defined in the **Variables** drop-down list in the **Channels** section. For instance, a simple formula in which the values from Vector 1 are subtracted from the values in Vector2 could be defined as "IV2 - IV1".

### Calculation Modes

This rule has two main modes: calculating rolling demand for a usage subscription based on either a rolling consumption total (Rolled Total) or rolling consumption average (Rolled Average), or calculating the coincident peak among all of the service points on a usage subscription based on either a rolling consumption total (Rolled Total - Coincident Peak) or rolling consumption average (Rolled Average - Coincident Peak). The coincident peak calculation can be performed on either all readings, or those designated as "on-peak" or "off-peak" via time of use mapping using the Coincident On-Peak or Coincident Off-Peak function, See **Defining Time of Use Mapping Options** for more information about time of use mapping.

The number of intervals in the rolling total or rolling average is determined via rule configuration. The **Roll Interval Size** parameter (in the **Calculation Details** section) divided by the **Common Interval Size** parameter (in the **Interval Data Retrieval Parameters** section). For example, if **Roll Interval Size** was 01:00:00 (1 hour) and the **Common Interval Size** was 00:15:00 (15 minutes), the rolling total or average would be based on sets of 4 intervals (1 hour divided by 15 minutes).

**Rolled Demand Total**

The "Rolled Demand Total" **Roll Calculate Function** identifies the peak demand for the usage subscription based on a rolling consumption total.

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period. This table includes the following:

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points


Copyright © 2026, Oracle and/or its affiliates.

- **Rolling Total**: The rolling total of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling total (calculated by multiplying the running total by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule would identify the highest demand for the usage subscription during the calculation period (indicated in **bold**).

| Date/Time | Service Point 1 | Service Point 2 | Total | Rolling Total | Demand |
|---|---|---|---|---|---|
| 10/27/2022 12:00 PM | 12 | 11 | 23 | | |
| 10/27/2022 1:00 PM | 10 | 12 | 22 | | |
| 10/27/2022 2:00 PM | 11 | 13 | 24 | | |
| 10/27/2022 3:00 PM | 13 | 13 | 26 | 95 | 95 |
| 10/27/2022 4:00 PM | 14 | 14 | 28 | 100 | 100 |
| 10/27/2022 5:00 PM | 13 | 13 | 26 | 104 | 104 |
| **10/27/2022 6:00 PM** | 12 | 13 | 25 | 105 | **105** |
| 10/27/2022 7:00 PM | 11 | 12 | 23 | 102 | 102 |
| 10/27/2022 8:00 PM | 12 | 13 | 25 | 99 | 99 |
| 10/27/2022 9:00 PM | 13 | 14 | 27 | 100 | 100 |
| 10/27/2022 10:00 PM | 12 | 14 | 26 | 101 | 101 |
| 10/27/2022 11:00 PM | 12 | 12 | 24 | 102 | 102 |

This rule would return the following values in the usage transaction service quantities list, which represents the peak demand value during the calculation period:

| Date/Time | Demand |
|---|---|
| 10/27/2022 6:00 PM | 105 |

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

**Rolled Demand Total - Coincident Peak / Rolled Demand Total - Coincident On Peak / Rolled Demand Total - Coincident Off Peak**


Copyright © 2026, Oracle and/or its affiliates.

The "Rolled Demand Total - Coincident Peak / Rolled Demand Total - Coincident On Peak / Rolled Demand Total - Coincident Off Peak" **Roll Calculate Function** identify the coincident peak demand for all of the service points on the usage subscription. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point). The coincident peak demand is the demand based on the rolling total at the coincident peak.

The data used in the calculation is based on the specific **Roll Calculate Function** selected:

- **Total - Coincident Peak**: Evaluates all measurements matching the configured unit of measure
- **Total - Coincident On Peak**: Evaluates measurements matching the configured unit of measure that falls with the "on peak" TOU period defined in the **Results** section based on a specified TOU Map Template.
- **Total - Coincident Off Peak**: Evaluates measurements matching the configured unit of measure that falls with the "off peak" TOU period defined in the **Results** section based on a specified TOU Map Template.

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period. This table includes the following:

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points
- **Rolling Total**: The rolling total of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling total (calculated by multiplying the running total by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule would identify the highest total for all of the service points within the calculation period (indicated in **bold**).

| Date/Time | Service Point 1 | Service Point 2 | Total | Rolling Total | Demand |
|---|---|---|---|---|---|
| 10/27/2022 12:00 PM | 12 | 11 | 23 | | |
| 10/27/2022 1:00 PM | 10 | 12 | 22 | | |
| 10/27/2022 2:00 PM | 11 | 13 | 24 | | |
| 10/27/2022 3:00 PM | 13 | 13 | 26 | 95 | 95 |
| **10/27/2022 4:00 PM** | 14 | 14 | 28 | 100 | **100** |
| 10/27/2022 5:00 PM | 13 | 13 | 26 | 104 | 104 |
| 10/27/2022 6:00 PM | 12 | 13 | 25 | 105 | 105 |
| 10/27/2022 7:00 PM | 11 | 12 | 23 | 102 | 102 |


Copyright © 2026, Oracle and/or its affiliates.

| Date/Time | Service Point 1 | Service Point 2 | Total | Rolling Total | Demand |
|---|---|---|---|---|---|
| 10/27/2022 8:00 PM | 12 | 13 | 25 | 99 | 99 |
| 10/27/2022 9:00 PM | 13 | 14 | 27 | 100 | 100 |
| 10/27/2022 10:00 PM | 12 | 14 | 26 | 101 | 101 |
| 10/27/2022 11:00 PM | 12 | 12 | 24 | 102 | 102 |

This rule would return the following values in the usage transaction service quantities list, which represent the demand at the coincident peak:

| Date/Time | Coincident Peak Demand |
|---|---|
| 10/27/2022 4:00 PM | 100 |

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

### Rolled Demand Average

The "Rolled Demand Average" **Roll Calculate Function** identifies the peak demand for the usage subscription based on a rolling average (calculated from the rolling total).

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period. This table includes the following:

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points
- **Rolling Total**: The rolling total of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Rolling Average**: The rolling average of 4 total values (based on a **Roll Interval Size** of 01:00:00 (1 hour) and **Common Interval Size** of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling average (calculated by multiplying the rolling average by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule would identify the highest demand for the usage subscription during the calculation period (indicated in **bold**).

| Date/Time | Service Point 1 | Service Point 2 | Total | Rolling Total | Rolling Average | Demand |
|---|---|---|---|---|---|---|
| 10/27/2022 12:00 PM | 12 | 11 | 23 | | | |


Copyright © 2026, Oracle and/or its affiliates.

| Date/Time | Service Point 1 | Service Point 2 | Total | Rolling Total | Rolling Average | Demand |
|---|---|---|---|---|---|---|
| 10/27/2022 1:00 PM | 10 | 12 | 22 | | | |
| 10/27/2022 2:00 PM | 11 | 13 | 24 | | | |
| 10/27/2022 3:00 PM | 13 | 13 | 26 | 95 | 23.75 | 23.75 |
| 10/27/2022 4:00 PM | 14 | 14 | 28 | 100 | 25 | 25 |
| 10/27/2022 5:00 PM | 13 | 13 | 26 | 104 | 26 | 26 |
| **10/27/2022 6:00 PM** | 12 | 13 | 25 | 105 | 26.25 | **26.25** |
| 10/27/2022 7:00 PM | 11 | 12 | 23 | 102 | 25.5 | 25.5 |
| 10/27/2022 8:00 PM | 12 | 13 | 25 | 99 | 24.75 | 24.75 |
| 10/27/2022 9:00 PM | 13 | 14 | 27 | 100 | 25 | 25 |
| 10/27/2022 10:00 PM | 12 | 14 | 26 | 101 | 25.25 | 25.25 |
| 10/27/2022 11:00 PM | 12 | 12 | 24 | 102 | 25.5 | 25.5 |

This rule would return the following values in the usage transaction service quantities list, which represents the peak demand value during the calculation period:

| Date/Time | Demand |
|---|---|
| 10/27/2022 6:00 PM | 26.25 |

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

**Rolled Demand Average- Coincident Peak /Rolled Demand Average-Coincident On Peak / Rolled Demand Average - Coincident Off Peak**

The "Average -Coincident Peak / Average - Coincident On-Peak / Average - Coincident Off-Peak"**Calculate Function**identify the coincident peak demand for all of the service points on the usage subscription. The "coincident peak" is the time at which the combined values of multiple service points is the highest (which is not necessarily the peak for each individual service point). The coincident peak demand is the demand based on the rolling average at the coincident peak.

The data used in the calculation is based on the specific**Roll Calculate Function**selected:

- **Rolled Demand Total - Coincident Peak**: Evaluates all measurements matching the configured unit of measure


Copyright © 2026, Oracle and/or its affiliates.

- **Rolled Demand Total - Coincident On Peak**: Evaluates measurements matching the configured unit of measure that falls with the "on peak" TOU period defined in the**Results**section based on a specified TOU Map Template.
- **Rolled Demand Total - Coincident Off Peak**: Evaluates measurements matching the configured unit of measure that falls with the "off peak" TOU period defined in the**Results**section based on a specified TOU Map Template.

For example, the table below lists sample hourly interval data for two service points belonging to single usage subscription for a 12 hour calculation period.This table includes the following:

- **Date/Time**: The date and time of each interval
- **Service Point n**: The kWh value for each interval at a specific service point
- **Total**: The combined total of consumption at the usage subscription's service points
- **Rolling Total**: The rolling total of 4 total values (based on a**Roll Interval Size**of 01:00:00 (1 hour) and**Common Interval Size**of 00:15:00 (15 minutes))
- **Rolling Average**: The rolling average of 4 total values (based on a**Roll Interval Size**of 01:00:00 (1 hour) and**Common Interval Size**of 00:15:00 (15 minutes))
- **Demand**: The demand calculated from the rolling average (calculated by multiplying the rolling average by the number of intervals per hour - in this example the total is the same as the demand because this is hourly data).

This rule wouldidentify the highest demand for the usage subscription during the calculation period (indicated in**bold**).

| Date/Time | Service Point 1 | Service Point 2 | Total | Rolling Total | Rolling Average | Demand |
|---|---|---|---|---|---|---|
| 10/27/2022 12:00 PM | 12 | 11 | 23 | | | |
| 10/27/2022 1:00 PM | 10 | 12 | 22 | | | |
| 10/27/2022 2:00 PM | 11 | 13 | 24 | | | |
| 10/27/2022 3:00 PM | 13 | 13 | 26 | 95 | 23.75 | 23.75 |
| **10/27/2022 4:00 PM** | 14 | 14 | 28 | 100 | 25 | **25** |
| 10/27/2022 5:00 PM | 13 | 13 | 26 | 104 | 26 | 26 |
| 10/27/2022 6:00 PM | 12 | 13 | 25 | 105 | 26.25 | 26.25 |
| 10/27/2022 7:00 PM | 11 | 12 | 23 | 102 | 25.5 | 25.5 |
| 10/27/2022 8:00 PM | 12 | 13 | 25 | 99 | 24.75 | 24.75 |
| 10/27/2022 9:00 PM | 13 | 14 | 27 | 100 | 25 | 25 |
| 10/27/2022 10:00 PM | 12 | 14 | 26 | 101 | 25.25 | 25.25 |
| 10/27/2022 11:00 PM | 12 | 12 | 24 | 102 | 25.5 | 25.5 |


Copyright © 2026, Oracle and/or its affiliates.

This rule would return the following values in the usage transaction service quantities list, which represents the peak demand value during the calculation period:

| Date/Time | Demand |
|---|---|
| 10/27/2022 4:00 PM | 25 |

> **Note**
>
> This option should be used for information and reporting purposes only. These values cannot be used for billing purposes.

### Notes and Examples

### Notes and Recommendations

- When using he "Coincident On Peak" or "Coincident Off Peak"**Roll Calculate Function**, you must specify either a TOU map (in the**TOU Map**field) or a TOU map template (in the**On Peak Off Peak Schedule**drop-down list) in the**Calculation Details**section,and specify the appropriate time of use period (On-Peak or Off-Peak, respectively) in the**TOU**field in the**Results**section.
- A recommended practice is to configure a Service Quantity Identifier (SQI) to uniquely identify coincident peak data and specify it in the**SQI**field in the**Results**section.
- Additional detail on the logic of this rule can be found in the **Detailed Description** of the D1-CALROLDEM Algorithm Type.

For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

### Business Object

D1-CalcRollingDemandRule

### Example Scenarios

Below are some example scenarios that can be achieved based on configuration of this rule.

**Scenario 1**: Calculate demand for services points based on rolling total (based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 00:15:00
  - **Channels**:
    - **Variable 1**:
      - **Variable Name**: Variable 1 (IV1)
      - **UOM**: Kilowatt-Hour
  - **Aggregation Method**:
    - **Formula Source**: Simple Vector Formula


Copyright © 2026, Oracle and/or its affiliates.

        - **Simple Vector Formula**: IV1
- **Calculation Details**:
    - **Roll Calculate Function**: Rolled Demand Total
    - **Roll Interval Size**: 01:00:00
- **Results**:
    - **UOM**: Kilowatt
    - **TOU**:
    - **SQI**:

**Scenario 2**: Calculate demand for service points based on rolling total (based on two channels)

In this scenario, the service point has two channels, one for consumed energy, and one for generated energy (from a solar panel). If the consumed value is greater than the generated value, subtract the generated value from the consumed value. If the generated value is greater than the consumed value, use zero (0).

- **Interval Data Retrieval Parameters**:
    - **Check Missing Intervals**: Yes
    - **No Meter Condition**: No Read - System
    - **Common UOM**: Kilowatt-Hour
    - **Common Interval Size**: 01:00:00
    - **Channels**:
        - **Variable 1**:
            - **Variable Name**: Variable 1 (IV1)
            - **UOM**: Kilowatt-Hour
            - **TOU**:
            - **SQI**: Consumed
        - **Variable 2**:
            - **Variable Name**: Variable 2 (IV2)
            - **UOM**: Kilowatt-Hour
            - **TOU**:
            - **SQI**: Generated
    - **Aggregation Method**:
        - **Formula Source**: Conditional Vector Formula
        - **Conditional Vector Formula**:
            - **Sequence**: 10
            - **Left Operand**: IV1
            - **Criteria Operator**: >
            - **Right Operand**: IV2
            - **True Action**: Apply True Formula
            - **True Formula**: IV1-IV2


Copyright © 2026, Oracle and/or its affiliates.

        - **False Action**: Apply False Formula
        - **False Formula**: 0
- **Calculation Details**:
  - **Roll Calculate Function**: Rolled Demand Total
  - **Roll Interval Size**: 01:00:00
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**:
  - **SQI**:

**Scenario 3**: Calculate demand for services points based on rolling average (based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 00:15:00
  - **Channels**:
    - **Variable 1**:
      - **Variable Name**: Variable 1 (IV1)
      - **UOM**: Kilowatt-Hour
  - **Aggregation Method**:
    - **Formula Source**: Simple Vector Formula
    - **Simple Vector Formula**: IV1
- **Calculation Details**:
  - **Roll Calculate Function**: Rolled Demand Average
  - **Roll Interval Size**: 01:00:00
- **Results**:
  - **UOM**: Kilowatt
  - **TOU**:
  - **SQI**:

**Scenario 4**: Total - Coincident Peak (based on a single channel)

- **Interval Data Retrieval Parameters**:
  - **Check Missing Intervals**: Yes
  - **No Meter Condition**: No Read - System
  - **Common UOM**: Kilowatt-Hour
  - **Common Interval Size**: 00:15:00
  - **Channels**:
    - **Variable 1**:


Copyright © 2026, Oracle and/or its affiliates.

        - **Variable Name**: Variable 1 (IV1)
        - **UOM**: Kilowatt-Hour
    - **Aggregation Method**:
        - **Formula Source**: Simple Vector Formula
        - **Simple Vector Formula**: IV1
- **Calculation Details**:
    - **Roll Calculate Function**: Rolled Demand Total, Coincident Peak
    - **Roll Interval Size**: 01:00:00
- **Results**:
    - **UOM**: Kilowatt
    - **TOU**:
    - **SQI**: Coincident Peak

**Scenario 5**: Average - Coincident On Peak (based on a single channel)

- **Interval Data Retrieval Parameters**:
    - **Check Missing Intervals**: Yes
    - **No Meter Condition**: No Read - System
    - **Common UOM**: Kilowatt-Hour
    - **Common Interval Size**: 00:15:00
    - **Channels**:
        - **Variable 1**:
            - **Variable Name**: Variable 1 (IV1)
            - **UOM**: Kilowatt-Hour
    - **Aggregation Method**:
        - **Formula Source**: Simple Vector Formula
        - **Simple Vector Formula**: IV1
- **Calculation Details**:
    - **Roll Calculate Function**: Rolled Demand Average, Coincident, On Peak
    - **Roll Interval Size**: 01:00:00
    - **Reference TOU Map by Template**: Yes
    - **On Peak Off Peak Schedule**: <TOU map template>
- **Results**:
    - **UOM**: Kilowatt
    - **TOU**: On Peak
    - **SQI**: Coincident On Peak

## Round and Adjust Usage

This rule copies identified source and target Service Quantities and inserts these as period Service Quantities that are rounded and adjusted. This rule captures the source and target


Copyright © 2026, Oracle and/or its affiliates.

Service Quantity identifiers, rounding method, and Service Quantity bucket to hold the adjustment. The usage calculation rule also has an option to validate the difference between the source and target service quantities.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-RNDADJUSG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-RoundAndAdjustUsage

## Vector and Service Quantity Math

This rule is designed to facilitate configuration of complex vector and scalar quantity calculations. It is based on a series of underlying services, including Axis Conversion, Apply Formula (most importantly), and Apply TOU Map. It can be configured to accept as input up to five vectors - where a single vector can represent the combination of all measuring components with linkages to the Usage Subscription with the configured UOM/SQI. It can also accept a list of scalar variables that can be taken in as factor values, usage transaction service quantities, set functions on a vector, or numeric values.

Vectors can be combined using a simple formula expression, or using condition formula expressions. Both interval values and interval conditions can be referenced. The result is interval-by-interval processing of the vector formula. The resulting final vector can be stored, TOU mapped, and/or subjected to a final set function (such as sum or max).

This rule stores service quantities of type "other" - meaning that the service quantities created by this rule do not have linkages to a specific Service Point or measuring component, given the potentially diverse sources of the data taken as input by this rule that lead to the final quantity.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-MATH Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-Math

**Vector and Service Quantity Math Usage Calculation Rules and Anchor Time**

The Vector and Service Quantity Math usage calculation rule supports usage calculations based on an "Anchor Time". The Anchor Time is specified on certain measuring component types and defines the start and end time of the business day for purposes of processing measurement data and usage transactions. When processing usage calculations for


Copyright © 2026, Oracle and/or its affiliates.

measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the measuring component type.

## Post-Calculation Usage Calculation Rules

- Usage Calculation High/Low Rule
- Validate Against Tolerance
- Business Flag Hold

### Usage Calculation High/Low Rule

This rule is used to validate the current usage against historical usage - previous year usage or previous usage. It ensures that any increase or decrease of the current usage relative to historical usage is within a tolerance.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-USGHIGLOW Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-UsageHighLowRule

### Validate Against Tolerance

This rule is used to validate the calculated usage against a tolerance value. The tolerance value may either come from the specified value or tolerance factor defined in the usage calculation rule.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-VALUSGTOL Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ValAgainstTol

### Business Flag Hold


Copyright © 2026, Oracle and/or its affiliates.

This rule can be used to place a hold on a usage transaction until the business flags related to the usage transactions service points can be investigated. The hold can be placed in one of two ways:

- **Calculation Window Based**: holds will remain until either a user manually bypasses the usage exception error or the current date is within a configurable tolerance of the calculation window end (aka the retry until date time).
- **Indefinite**: holds require a user to manually bypass the usage exception that is generated.

Defining which business flags should result in a usage transaction is a matter of identifying which business flag types, priorities, and confidences should be included.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-AUR-IBFH algorithm type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-BusinessFlagHold

## Decision-Making Usage Calculation Rules


- Execute Usage Calculation Group
- Exception Handler


### Execute Usage Calculation Group

This rule performs a call to execute a separate usage calculation group which includes execution of all usage calculation rules within that group. It is especially useful in cases where repeating the same rule over and over would be hard to maintain.

For example, you may want to calculate a straightforward kWh sum for every usage calculation group. Creating a separate CALC_KWH_SUM usage calculation group for this one calculation allows you to isolate the configuration points. Then the CALC_KWH_SUM usage calculation group can be referenced in every other usage calculation group that needs this sort of service quantity.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-AUSGR-RFG Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**



Copyright © 2026, Oracle and/or its affiliates.

D1-UsageRuleReferredUsageGroup

### Exception Handler

This rule evaluates the exception list that was accumulated during the execution of the calculation rules against exception criteria configured on this rule. This used to terminate the execution of usage calculation group processor if there are far too many exceptions hit during the execution.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-UREXPCHAN Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-UsageRuleExceptionHandler

## Detailed Configuration Examples

**Demand Calculation Options Using Interval Data**

There are a number of ways to calculate a demand value from interval data when retrieving bill determinants for CIS. A few examples of demand calculations are listed below:

**Scenario1: Use interval data for demand at a common block size**: the Vector and Service Quantity Math usage calculation rule can be used to calculate demand from the interval data. Below is one example of configuration to handle demand calculation based on 30 minute blocks:

| Usage Calculation Rule field | Value | Notes |
|---|---|---|
| Vector 1 // Type | Physical Channels Links to Usage Subscription | |
| Vector 1 // Unit of Measure | Kilowatt-Hours | Use your own UOM here - this is just as example. |
| Vector 1 // Service Quantity Identifier | Consumed | Use your own SQI here - this is just as example. |
| Vector Processing // Common Interval Size | 00:30:00 | This configuration rolls up all channels (whether 5 minute, 15 minute, or 30 minute interval data) into a common interval size. |
| Vector Processing // Vector Formula Source | Simple Vector Formula | |
| Vector Processing // Simple Vector Formula | IV1*2 | This is multiplying the common 30 minute interval size by 2 to achieve an hourly value. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Result // Unit of Measure | Kilowatts | Use your own UOM here - this is just as example. |
| Result // Service Quantity Identifier | Consumed | Use your own SQI here - this is just as example. |
| Result // Insert Primary SQ Entry | Yes | |
| Result // SQ Entry Quantity Source | Set Function Against Derived Vector | |
| Result // Set Function Against Derived Vector | Max | This function pulls the max value based on the configuration in the "Vector Processing" section. |

**Scenario 2: Use interval data for demand by TOU period**: This example is very similar to the last and again leverages the Vector and Service Quantity Math usage calculation rule. However, instead of calculating a single demand value for the entire period it will calculate a demand value for each TOU period:

| Usage Calculation Rule field | Value | Notes |
|---|---|---|
| Vector 1 // Type | Physical Channels Links to Usage Subscription | |
| Vector 1 // Unit of Measure | Kilowatt-Hours | Use your own UOM here - this is just as example. |
| Vector 1 // Service Quantity Identifier | Consumed | Use your own SQI here - this is just as example. |
| Vector Processing // Common Interval Size | 00:30:00 | This configuration rolls up all channels (whether 5 minute, 15 minute, or 30 minute interval data) into a common interval size. |
| Vector Processing // Vector Formula Source | Simple Vector Formula | |
| Vector Processing // Simple Vector Formula | IV1*2 | This is multiplying the common 30 minute interval size by 2 to achieve an hourly value. |
| Result // Unit of Measure | Kilowatts | Use your own UOM here - this is just as example. |
| Result // Service Quantity Identifier | Consumed | Use your own SQI here - this is just as example. |
| Result // Insert Primary SQ Entry | No | |
| Result // Apply TOU Map To Derived Vector | Yes | |
| Result // TOU Map | (your TOU Map) | Select the TOU Map you'd like to apply. |
| Result // Time Of Use Calculate Function | Max | This function pulls the max value based on the configuration in the "Vector Processing" section and then buckets it based on the selected TOU Map. |


Copyright © 2026, Oracle and/or its affiliates.

# Usage Calculation Types


- Understanding Usage Calculation Types
- Configuring Usage Calculation Types


## Understanding Usage Calculation Types

Usage calculation types define calculations executed at a customer level where there may be more than one calculation per usage subscription. For example, a customer may have a usage subscription for producing billing determinants but there may be additional calculations that will be executed on a much less frequent basis (annually, quarterly, etc). Usage calculation types provide a way to leverage a single usage subscription to drive multiple different calculations.

Usage calculation types use the following parameters:

- **Request Type**: Used to drive the creation and processing of the usage transactions for all usage subscriptions that are valid for the usage calculation type.

  > **Note**
  >
  > Request types for usage calculation types should be based on the Usage Calculation Request Type business object (D2–UsageCalculationReqType)

- **Service Point Quantity Creation**: Details of the quantities created by usage calculations of this type, including service point quantity type, and UOM, TOU, and SQI.
- **Applicable Usage Subscription Types**: A list of usage subscription types for which this usage calculation type is applicable. Each usage subscription type is defined by an effective date range, the usage transaction business object used for the usage calculation, the result UOM/TOU/SQI, and the usage calculation group used to perform the calculations.

## Configuring Usage Calculation Types

You use the **Usage Calculation Type** portal to maintain and view usage calculation types.

Refer to Understanding Usage Calculation Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Calculation Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Calculation Type List**: This zone lists all usage calculation type records. Broadcast a record to display the details of the selected record.
- **Usage Calculation Type**: This zone displays details for the selected usage calculation type.

# Usage Subscription Quantity Types



Copyright © 2026, Oracle and/or its affiliates.

- Understanding Usage Subscription Quantity Types
- Configuring Usage Subscription Quantity Types


## Understanding Usage Subscription Quantity Types

Usage subscription quantity types define types of quantities that can be stored for a usage subscription. These quantity types are often infrequently calculated descriptors of a given customer and how they relate, at an aggregate level, to either the customer base as a whole or their particular rate class. For example, a customer quantity may be used to store a scaling factor that describes how a given customer's usage compares to the profiled usage for their rate class.

In a settlement implementation, usage subscription quantity types can be used to define values calculated monthly, annually, etc., such as annual peak load contribution (PLC).

Usage subscription quantity types use the following parameters:

- **Service Type**: The service type (electric, gas, water, etc.) for service point quantities of this type.
- **Usage Subscription Quantity Business Object**: The business object used for usage subscription quantities of this type.
- **Usage Subscription Quantity Identifiers**: Value identifiers related to the current usage subscription quantity type (used to provide shorthand descriptions of the various types of values measured by usage subscription quantities of this type).

## Configuring Usage Subscription Quantity Types

You use the **Usage Subscription Quantity Type** portal to display and maintain usage subscription quantity types.

Refer to Understanding Usage Subscription Quantity Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Usage Subscription Quantity Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Usage Subscription Quantity Type List**: This zone lists all usage subscription quantity type records. Broadcast a record to display the details of the selected record.
- **Usage Subscription Quantity Type**: This zone displays details for the selected usage subscription quantity type.

# Dynamic Option Types


- Understanding Dynamic Option Types
- Configuring Dynamic Option Types


## Understanding Dynamic Option Types



Copyright © 2026, Oracle and/or its affiliates.

Dynamic options specify terms that override how usage is normally calculated, such as a critical peak period that affects the TOU mapping of interval consumption. Dynamic option types store information common to dynamic options of a specific type.

There are two classes of dynamic options:

- **Interval Set**: Used to represent dynamic option types that provide a set of intervals that will be included in a calculation.
- **Usage Event**: Used to represent dynamic option types that provide a specific period of time during which a dynamic option is applicable.

Interval Set dynamic option types also define key information about the interval data they represent, including:

- **Unit of Measure**: Identifies the quantity that Interval Set dynamic options and Interval Set dynamic option events based on this type are measuring.
- **Interval Size**: Defines the size of the intervals for Interval Set dynamic option events of this type, represented as hours:minutes:seconds (HH:MI:SS).

## Configuring Dynamic Option Types

This portal is used to display and maintain a Dynamic Option Type.

Refer to Understanding Dynamic Option Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Dynamic Option Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Dynamic Option Type List**: displays all of the Dynamic Option Types so you can choose the one you want to display in more detail
- **Dynamic Option Type**: shows the specific configuration for the selected Dynamic Option Type

# Contact Types

- Understanding Contact Types
- Configuring Contact Types

## Understanding Contact Types

Contact types define the properties of a class of entities (businesses, persons).

Contact types can be created using the following Contact Business Objects:

- **Business** (D1-Business): Used for business contacts. Cannot be changed.
- **Person** (D1-Person): Used for person contacts. Cannot be changed.
- **Contact** (D1-Contact): Can be used for Business or Person contacts. Use the **Person/Business** drop-down to specify Business or Person. This business object allows for contacts to be changed from Business to Person, or Person to Business if needed.


Copyright © 2026, Oracle and/or its affiliates.

## Configuring Contact Types

This portal is used to display and maintain a Contact Type.

Refer to Understanding Contact Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Contact Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Contact Type List**: displays all of the Contact Types so you can choose the one you want to display in more detail
- **Contact Type**: shows the specific configuration for the selected Contact Type

# Bill Cycle

- Understanding Bill Cycles
- Configuring Bill Cycles

## Understanding Bill Cycles

Bill cycles identify periods when a customer is going to be billed and when bill determinants are calculated.

## Configuring Bill Cycles

The **Bill Cycle** portal is used to display and maintain bill cycles.

Refer to Understanding Bill Cycle for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **Bill Cycle**.

The following zones may appear as part of the portal's **Main** tab page:

- **Bill Cycle List**: displays all of the Bill Cycles so you can choose the one you want to display in more detail.
- **Bill Cycle**: shows the specific configuration for the selected Bill Cycle.
- **Bill Cycle List**: lists the bill cycle schedules of the current bill cycle.

# Defining Time of Use Mapping Options

- Time of Use Groups
- Time of Use Map Templates
- Time of Use Map Types

# Time of Use Groups


Copyright © 2026, Oracle and/or its affiliates.

- Understanding Time of Use Groups
- Configuring Time of Use Groups

## Understanding Time of Use Groups

Time of Use (TOU) Groups are groups of TOUs that are associated to TOU templates that limit the TOUs available for use in a TOU schedule. These TOUs will be available as the default, holiday, and schedule TOUs within the template.

Each TOU in the group is given a priority. The priority is important in a few ways:

- The TOU Overlay 360 Degree zone uses these priorities to decide which TOUs will be rendered as a unique shade and which will fall into a category of "other" when the maximum number of distinct TOUs to graph has been reached.
- These priorities are available for use in customized logic.

## Configuring Time of Use Groups

This portal is used to display and maintain a TOU Group.

Refer to Understanding Time of Use Groups for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **TOU Group**.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Group List:** This zone lists all TOU Group records. Broadcast a record to display the details of the selected record.
- **TOU Group:** This zone provides information about the selected TOU Group.

# Time of Use Map Templates

- Understanding Time of Use Map Templates
- Configuring Time of Use Map Templates

## Understanding Time of Use Map Templates

Every TOU map references a TOU map template that defines the rules for generating TOU data from that TOU map. Specifically, TOU map templates define:

- The TOU group (defines the valid TOU periods for the template) used for the TOU map
- The default TOU period used for periods not explicitly defined. (This means you don't have to specify dates and times for all periods. For example, if your default TOU period is "Off Peak" you only need to define dates and days and times for On Peak or other TOU periods.)
- The specific date ranges, days of the week, and time periods designated for each TOU period.

The system periodically generates TOU map data for TOU maps by interpreting the rules defined template.


Copyright © 2026, Oracle and/or its affiliates.

Attributes used to define TOU map templates include the following:

- **TOU Group**: the TOU group used by the map template Default TOU: the default TOU for the map template (from the TOU Group). This is the TOU used when creating TOU map data for dates not accounted for in the TOU Schedules section.
- **Work Calendar**: the work calendar used to identify holidays. On each holiday the Holiday TOU will be used. For additional information see Defining Work Calendar in the Oracle Utilities Application Framework *Administrative User Guide*.
- **Holiday TOU**: the TOU used for holidays (from the TOU Group)
- **Holiday Template**: the TOU map template used for holidays (if applicable)
- **Interval Size**: the size of the intervals for TOU map data created from the map template, represented as hours:minutes:seconds (HH:MI:SS).
- **TOU Schedules**: date ranges (including month, day, and time ranges) and which TOU periods apply to each

Refer to Understanding Referencing Master Data by Identifiers for information on how admin configuration can reference a TOU map by a TOU template to ease migration of data between environments.

### TOU Map Template Interval Size

TOU map templates can also specify an interval size (in seconds-per-interval, or SPI). This value specifies the duration of the individual TOU map data records, and also controls the values allowed in the Start and End Times. For example, if a TOU map template sets the interval size at 15 minutes, Start and End times must be in units of the interval size (10:00, 10:15, 10:30, etc.).

A TOU map template can be used to generate TOU map data for TOU maps whose SPI is divisible by the template's SPI. For example, a 60 minute template can be used to generate TOU data for TOU maps with SPIs of 60 minutes, 15 minutes, 5 minutes, etc. This means separate map templates are not needed for every SPI.

### Holidays

Many utilities categorize consumption on holidays differently than on the day of week on which the holiday falls. For example, holiday consumption might be categorized as Off-Peak regardless of the day it falls on. TOU map templates can define rules for different TOU periods for holidays in two ways. Both options require that the template references a Work Calendar that identifies each of the holidays throughout the year. In addition they require either:

- A Holiday TOU that will be used for each holiday for the duration of that holiday (e.g. Off Peak)
- A Holiday TOU Map Template that defines the TOUs that should be used for holidays that fall during different seasons within the year (e.g. Off Peak Summer, Off Peak Winter)

### Important Time of Use Template System Events

The time of use template supports the following business object algorithm system events:

- **Derive TOU**: receives a date time and determines the TOU code for that date time based on the configuration of the time of use template schedule. See the Derive Time Of Use For Date Time (D2-DERTOU-DT) algorithm type as an example.

# Configuring Time of Use Map Templates


Copyright © 2026, Oracle and/or its affiliates.

This portal is used to display and maintain a TOU Map Templates.

Refer to Understanding Time of Use Map Templates for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **TOU Map Template**.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Map Template List:** This zone lists all TOU Map Template records. Broadcast a record to display the details of the selected record.
- **TOU Map Template:** This zone provides information about the selected TOU Map Template.

# Time of Use Map Types

- Understanding Time of Use Map Types
- Configuring Time of Use Map Types

## Understanding Time of Use Map Types

TOU Map Types define important properties of TOU maps of the type, including the interval size and the valid TOU map templates.

Attributes used to define TOU map types include the following:

- **Time Zone**: the time zone to be used when generating the TOU map data. Refer to Multiple Time Zone Support for more information about how time zones impact TOU map data in multiple time zone environments.
- **Interval Size**: the size of the intervals for TOU map data created from maps of this type, represented as hours:minutes:seconds (HH:MI:SS). The interval size cannot be larger than the interval size defined on the Default TOU Map Template or any of the Override TOU Map Templates.
- **Default TOU Map Template**: the default TOU map template used by maps of this type
- **Override TOU Map Templates**: one or more TOU map templates that can be used as an override on TOU maps of this type.

### TOU Map Type Interval Size

The SPI of a TOU map must divide evenly into the SPI of any measuring component that uses the map (because the system joins the date/time of the measurement to the date/time of the TOU data). This means that it is possible to use a 15 minute TOU map with a 60 minute measuring component. However, it is not OK to have a 60 minute TOU map used with a 15 minute measuring component because the join will miss 3 out of 4 measurements.

However, it is important to note that the TOU mapping process is at its most efficient when the measurement data that is being mapped is of the same interval size as the TOU schedule it is being mapped against. When there are differences in the interval size the process must first convert the measurement data into the appropriate interval size for the TOU Map prior to applying the TOU Map.

This means that for each TOU Map Template you should have sufficient TOU Map Types to cover the various interval sizes that will be supported by your measurement data. For example, if you have measuring components with interval sizes of 15 minutes, 30 minutes, and 60


Copyright © 2026, Oracle and/or its affiliates.

minutes then for each TOU Map Template there should be TOU Map Types with interval sizes of 15 minutes, 30 minutes, and 60 minutes.

**Default and Override TOU Map Templates**

While most TOU maps will use the TOU map template defined on the TOU map type, TOU maps also support a fallback/override pattern used in other areas of the system.

- A TOU map's TOU map type defines the default (or "fallback") TOU map template that's used to generate its TOU data.
- A TOU map's type defines the TOU map templates that can be referenced on individual TOU maps to override the "fallback" template.
- An individual TOU map can have an override template. If the TOU map doesn't have an override template, the fallback template defined on the TOU map type is used to generate the map's TOU data.

**Important Time of Use Map System Events**

The TOU Map business object that is associated to a given TOU map type supports a special system event that is used in the generation of TOU map data:

- **Create TOU Map Data**: receives a date range and for that date range it will create the appropriate TOU map data for the TOU map. See the Create TOU Map Data (D2-CRETMD-CT) algorithm type as an example.

## Configuring Time of Use Map Types

This portal is used to display and maintain a TOU Map Types.

Refer to Understanding Time of Use Map Types for more information.

You can access the portal by selecting **Admin**, then **Usage**, then **TOU Map Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **TOU Map Type List:** This zone lists all TOU Type records. Broadcast a record to display the details of the selected record.
- **TOU Map Type:** This zone provides information about the selected TOU Map Type.

# Defining Communications Options


- Device Event Types
- Activity Types
- Communication Types
- Service Task Types


## Device Event Types


- Understanding Device Event Types
- Configuring Device Event Types




Copyright © 2026, Oracle and/or its affiliates.

# Understanding Device Event Types

Device event types define properties common to specific types of events.

Device event types represent different types of events that can take place relative to a device. Examples of device events include power outages, power restoration, tampering alerts, and other events.

Device event types can be defined by the following attributes:

- **Standard Event Name:** the "standard" name of the event type. Device vendors may have their own specific names for device events. Only a single active device event type may be mapped to a given standard name at any time.
- **Device Event Category:** a category (defined as an Extendable Lookup) used to group device event types.
- **Reporting Category:** a category used to group device event types for reporting purposes.
- **Activity Type:** the activity type for activities created for device events of this type.

### Device Event Mapping

The first step to device event type configuration is defining the list of standard event names that will be processed by the system. This is done by populating the Standard Event Name extendable lookup (D1-StdEventNameLookup). More information about this extendable lookup can be found in About Device Events.

With the fully defined list of Standard Event Names each head-end specific event name (which is also called an external event name) that the system will receive from a given head-end should be mapped to a standard event name. This mapping is configured using a device event mapping extendable lookup. Each head-end system should have its own extendable lookup to define event name mapping in order to prevent possible conflicts between mappings. Each of these business objects should be defined as a child of the parent business object Device Event Mapping (D1-DeviceEventMappingLookup).

For example, head-end systems A and B might both use the same event name, such as the code "1", but this event might need to be mapped to "outage" for head-end system A but "tamper" for head-end system B.

The device event mapping extendable lookup business object is configurable. Each Oracle Utilities Smart Grid Gateway adapter includes a device event mapping lookup business object for the supported head-end system.

Lastly, each device event type is associated to a standard name. This means that as each device event that is received will go through the following mapping steps:

1. The head end specific event name is mapped to a standard event name using the head end specific device event mapping extendable lookup
2. The standard event name is mapped to a device event type
3. The device event type is used to select the appropriate device event business object

### Device Event - Additional Processing

Certain device event business objects can be used to update master data as a result of device events received into the system. For example, when a utility's field worker arms a meter, the resulting device event can trigger an update to the device's arming status in the Oracle Utilities application. This processing is initiated by the Execute - Additional Processing system event


Copyright © 2026, Oracle and/or its affiliates.

algorithm during the Additional Processing status of the business object's lifecycle.

The following base package business objects are configured to execute additional processing:

If a standard device event requires additional processing, the algorithms that execute the processing should be specified in the Additional Processing Algorithms list on the Standard Event Name extendable lookup for the event. The base package includes the additional processing algorithm Arm Meter (D1-ARM-METER).

Only a subset of base package device event business objects are configured to execute additional processing

| Description | Business Object Name |
|---|---|
| Device Event Communication Response | D1-DeviceEventComResp |
| Standard Device Event | D1-StandardDeviceEvent |

If additional processing is required for any business objects not listed above it can easily be added by configuring the algorithm Executer - Additional Processing System Event (D1-EXTADDPRC) in the Additional Processing status.

### Reader Remarks

Reader remark types define properties common to specific types of reader remarks.

Reader remarks are a type of device event used to capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters. Reader remark types represent the different types of remarks that meter readers can record. Examples of reader remark types include evidence of tampering, broken seals, damaged meter, dog on premises, and other notices.

When creating a new device type there are the following options:

| Name | Details | Business Object |
|---|---|---|
| Reader Remark Type | Provides the configuration for a reader remark type | D1-ReaderRemarkType |

### Field Activity Remarks

Field activity remark types define properties common to specific types of field activity remarks.

Field activity remarks are a type of device event used to capture and/or record specific events or circumstances encountered when a field worker is performing field work at a service point. Field activity remarks represent can represent situations that are found as well as actions that were taken in the field.

When creating a new device type there are the following options:

| Name | Details | Business Object |
|---|---|---|
| Device Field Remark Type | Provides the configuration for a field activity remark type | D1-DeviceFieldRemarkType |

# Configuring Device Event Types

This portal is used to display and maintain a Device Event Type.


Copyright © 2026, Oracle and/or its affiliates.

Refer to Understanding Device Event Types for more information.

You can access the portal by selecting **Admin**, then **Communication**, then **Device Event Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Device Event Type List:** This zone lists all Device Event Type records. Broadcast a record to display the details of the selected record.
- **Device Event Type:** This zone provides information about the selected Device Event Type.

# Activity Types

- Understanding Activity Types
- Configuring Activity Types
- Configuring Service Order Management Activity Types

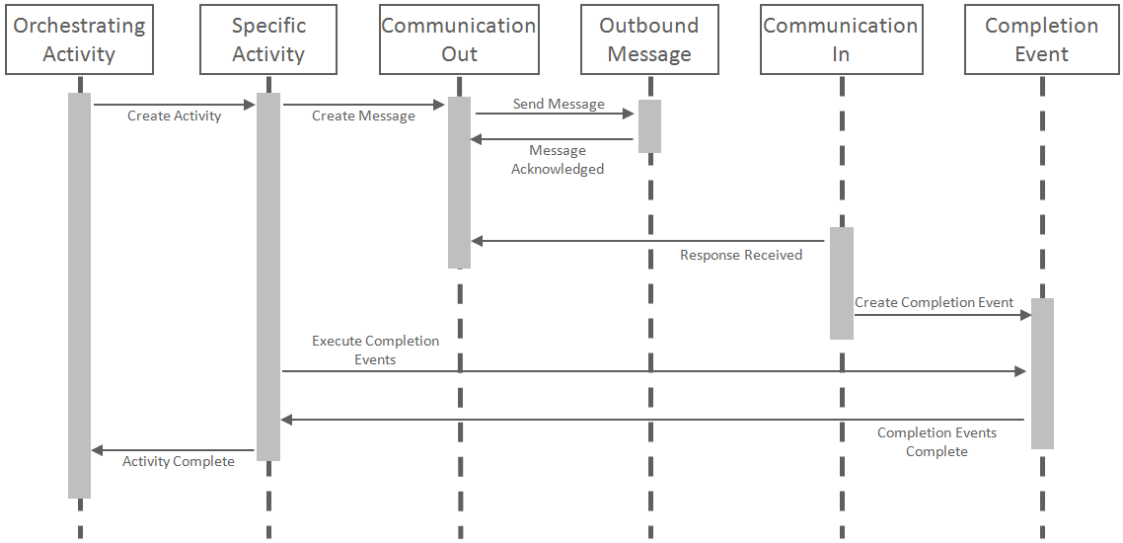
## Understanding Activity Types

Activities are records of a communication or event related to a device, measuring component, service point or other entity in the system. Examples of activities include smart meter command request, field activities, meter read downloads (for manually read meters) or the combined event of a "last gasp" and "power up" message sent by devices when they detects they experience an outage.

Each type of activity is assigned to an activity type category, please refer to About Activities for more information about the activity type categories supported by base product.

### Activities Orchestrate Communication

Certain types of activities such as request orchestrators, field activities, and command requests coordinate a large number of child transactions that represent the communication to and from an external application. The following diagram depicts a two-way communication with an external system:



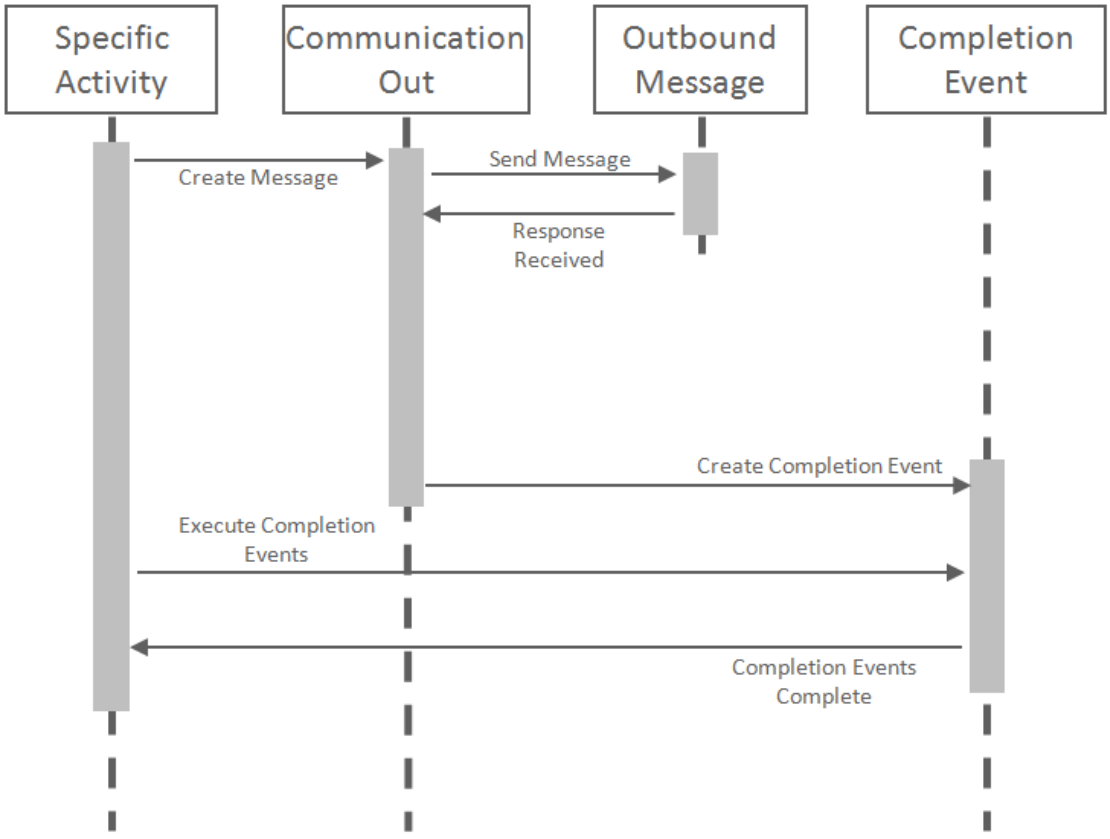


Copyright © 2026, Oracle and/or its affiliates.

Each object in the sequence diagram has a distinct set of duties within the context of the communication:

- Orchestrating Activity: controls the overall intent of the communication. For example it may "Start Service" which would include initiating one-to-many specific activities to install the meter and begin the flow of the metered commodity to the service point.
- Specific Activity: can be initiated from an orchestrating activity or directly. These activities represent a single task to be carried out such as installing a meter or a remote disconnect smart meter command.
- Communication Out: orchestrates the communication to the external application and provides robust handling for any errors that might occur during that communication.
- Outbound Message: represents the message payload sent to the external system and the synchronous response.
- Communication In: orchestrates the handling of an asynchronous or unsolicited response from an external system.
- Completion Event: carries out the results of the communication. For example, in the case of a remote connect it would create the appropriate on off history entry for the device's installation event.

The following diagram involves many of the same objects but instead represents a one-way communication with an external system. All objects maintain the same duties described above with the addition of the communication out handling the synchronous response which contains the result of the message.





Copyright © 2026, Oracle and/or its affiliates.

### Important Activity Type System Events

The activity type supports several business object algorithm system events that relate to calculating the consumption for measuring components of that type:

- **Customer-Device Compatibility**: receives information about the activity being analyzed and returns an indication of whether the customer and device are compatible. This system event is primarily used in service order management. See algorithm type Ensure Device-Usage Calculation Group Compatibility (D2-ENSDVCCMP) as an example.
- **Override Device / Task**: receives information about the activity such as whether the device is installed at a service point, whether the service point is connected, where it is disconnected (if applicable), and the installation event status override. Based on those inputs it should determine the output device configuration type as well as the field activity business object and field task type. This system event is primarily used in service order management. See algorithm type Evaluate Smart Meter Opt-Out for Device Installation (D2-EVSMOPDV) as an example.

### Service Order Activity and Service Order Field Activity Types and Divisions

Service order activity types and service order field activity types can specify a Division, to represent the specific operating company associated with activities of that type. If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", users can only access service order activities or service order field activities with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the**Miscellaneous** tab of the **User** portal.

> **Note**
>
> In Oracle Utilities Customer to Meter, the **Control By CIS Division** setting on the **Main** tab on the **Installation Options** portal is used to restrict data access based on divisions.

See Understanding Divisions for more information about divisions.

### Activity Type Maintenance Category

Activity types can be assigned an Activity Type Maintenance Category to specify that they should be maintained using the SOM Activity Type portal rather than the standard Activity Type portal. This option is used with the following categories of activity types:

- Field Activity Types
- Non-Dispatchable Activity Types
- Orchestration Maintenance Types
- Request Orchestration Types

Assigning activity types to the "Service Order Management Activity Type" Activity Type Maintenance Category is done via the Activity Type Maintenance Category business object option on the appropriate activity type business objects.

# Configuring Activity Types

You use the **Activity Type** portal is to view and maintain activity types.


Copyright © 2026, Oracle and/or its affiliates.

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured activity types as well activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type**: This zone provides information about the selected Activity Type
- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

  > **Note**
  >
  > If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

  > **Note**
  >
  > If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

# Configuring Service Order Management Activity Types


Copyright © 2026, Oracle and/or its affiliates.

You use the **SOM Activity Type** portal to view and maintain activity types whose business object has the Activity Type Maintenance Category business object option set to D1SO (Service Order Management Activity Type).

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **SOM Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured service order management activity types as well activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type**: This zone provides information about the selected Activity Type
- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

  > **Note**
  >
  > If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

  > **Note**
  >
  > If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.


Copyright © 2026, Oracle and/or its affiliates.

# Communication Types


- Understanding Communication Types
- Configuring Communication Types


## Understanding Communication Types

Communication types define properties common to a specific type of communication. This single admin object covers both communication in and communication out types.

Communication types include types of communications between an application and an external system such as a head-end system or the work management system. These are outbound communications such as smart meter command requests or field activities as well as inbound communications such as a message response that indicates the results of a request.

Communication types typically provide configuration around exception handling:

- **To Do Type and Role**: identifies the To Do to be created when an error is encountered
- **Retry Frequency**: defines how often the communication should be retried. As many errors can be due to connectivity the ability to retry provides automatic resolution.
- **Maximum Retries**: provides an upper limit to the number of times a communication can be re-attempted
- Each communication type can have its own unique set of fields, please refer to the embedded help for that communication type for more details on those fields.

## Configuring Communication Types

This portal is used to display and maintain a Communication Type.

Refer to Understanding Communication Types for more information.

You can access the portal by selecting **Admin**, then **Communication**, then **Communication Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Communication Type List:** This zone works differently than the typical zone that list types in that it displays both those communication types that have been configured as well as those communication types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Communication Type:** This zone provides information about the selected Communication Type

# Service Task Types


- Understanding Service Task Types
- Configuring Service Task Types




Copyright © 2026, Oracle and/or its affiliates.

## Understanding Service Task Types

Service tasks types define properties common to specific types of service tasks.

Service task types represent types of tasks that can be performed by users of other Oracle Utilities applications, such Oracle Utilities Customer Self Service or Oracle Utilities Network Management System. Examples of service tasks include self service meter reads, in which users enter their own meter reads via the Customer Self Service application, and service issue monitor types used when determining if service investigation is needed for a service point.

When creating a new record, the following options are available:

| Name | Details | Business Object |
|---|---|---|
| Service Issue Monitor Type | The service issue monitor type defines the parameters for when a service investigative order should be created. It can also optionally define the completion parameters for that indicate the investigative order was successful in its purpose. | D1-ServiceIssueMonitorType |
| Self-Service Rate Compare Scenario Request Type | A simple description of a rate compare scenario request type. | D2-RateCompareScenarioRqstType |
| Self-Service Meter Read Task Type | Provides the default settings for meter reads created through self service. | D2-SSMeterReadTaskType |

## Configuring Service Task Types

This portal is used to display and maintain a Service Task Type.

Refer to Understanding Service Task Types for more information.

You can access the portal by selecting **Admin**, then **Customer**, then **Service Task Type**

The following zones may appear as part of the portal's **Main** tab page:

- **Service Task Type List:** This zone lists all Service Task Type records. Broadcast a record to display the details of the selected record.
- **Service Task Type:** This zone provides information about the selected Service Task Type.

# Smart Grid Gateway Adapters

Oracle Utilities Smart Grid Gateway (SGG) Adapters support communication with various third-party head end systems to provide the following functionality:

- **Payload Processing**: Receiving usage and device event payloads from the head end system (also referred to as Measurement Data and Device Event Loading and Processing)
- **Smart Meter Command Processing**: Sending/receiving messages to/from third-party applications to initiate smart meter commands from Smart Grid Gateway


Copyright © 2026, Oracle and/or its affiliates.

### Smart Grid Gateway Implementations

Oracle Utilities Smart Grid Gateway Adapters can be implemented in one of two manners:

- **Middleware Implementations** use Oracle Service Bus (OSB) for payload processing and Oracle Business Process Execution Language (BPEL) to communicate with head end systems when initiating smart meter commands. Middleware implementations can be used **only** with on-premises implementations of Oracle Utilities Smart Grid Gateway.
- **Native Implementations** use batch processing for payload processing, and specific communication business objects to communicate with head end systems when initiating smart meter commands. Native implementations can be used with **either** on-premises implementations of Oracle Utilities Smart Grid Gateway, or with Oracle Utilities cloud services such as Oracle Utilities Meter Solution Cloud Service or Oracle Utilities Customer Cloud Service.

The following sections provide details about each type of implementation.


- Smart Grid Gateway Adapter Middleware Implementations
- Smart Grid Gateway Adapter Native Implementations


# Smart Grid Gateway Adapter Middleware Implementations

> **Note**
>
> Unless otherwise noted, the contents of this section apply to middleware (on-premises) implementations of Oracle Utilities Smart Grid Gateway only, and do NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.

Oracle Utilities Smart Grid Gateway (SGG) Adapters support communication with various third-party head-end systems. Smart Grid Gateway uses Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL) to facilitate communication with the head-end systems. Most if not all of the functionality shown below is available in each SGG Adapter.

**Measurement Data and Device Event Loading**: Data parsing and transformation via Oracle Service Bus from head-end system format into the Oracle Utilities Service and Measurement Data Foundation unified format for measurement data and device events.

**Measurement Data and Device Event Processing**: Configurable mapping for head-end system status codes and device event names to Oracle Utilities Service and Measurement Data Foundation standard values.

**Smart Meter Command Processing**: Sending/receiving messages to/from third-party applications to initiate smart meter commands from SGG. Most head-end systems support the following types of commands and communications:

- **Device Status Check:** Business objects and BPEL processes to support issuing device status check commands.
- **Meter Commissioning**: Business objects and BPEL processes to support issuing meter commissioning commands (including registration and installation commands).



Copyright © 2026, Oracle and/or its affiliates.

- **Meter Decommissioning**: Business objects and BPEL processes to support issuing meter decommissioning commands.
- **Meter Retirement:** Business objects and BPEL processes to support issuing meter retire (deregistration) commands.
- **On-Demand Read**: Business objects and BPEL processes to support issuing on-demand read commands.
- **Remote Connect**: Business objects and BPEL processes to support issuing remote connect commands.
- **Remote Disconnect**: Business objects and BPEL processes to support issuing remote disconnect commands.

This information in this guide describes how to configure, extend, and test the Smart Grid Gateway Adapters for individual head-end systems. The following head-end systems each have a specific SGG Adapter that can be configured: Itron OpenWay, Landis+Gyr, Networked Energy Services, MV90 for Itron, Sensus RNI, and Silver Spring Networks. Customers also can create their own customized adapter for using Smart Grid Gateway with a specific head-end system. See Creating a Custom Adapter for Smart Grid Gateway for more information.

- Configuring Smart Grid Gateway Adapters
- Using Smart Grid Gateway Middleware Test Harnesses
- Creating a Custom Adapter for Smart Grid Gateway

# Configuring Smart Grid Gateway Adapters

This section describes how to configure Oracle Utilities Smart Grid Gateway Adapters for each supported head-end system. Use the cross-references below to go to the adapter of interest.

Itron OpenWay

Landis+Gyr

Networked Energy Services

MV90 for Itron

Sensus RNI

Silver Spring Networks

> **Note**
>
> This section applies to middleware (on-premises) implementations of Oracle Utilities Smart Grid Gateway only. This section does NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.

- Itron OpenWay
- Landis+Gyr
- MV90 for Itron
- Networked Energy Services


Copyright © 2026, Oracle and/or its affiliates.

- Sensus
- Silver Spring Networks

## Itron OpenWay

The Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay supports communication with the Itron OpenWay application, including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, status check, and on-demand read commands. The following table describes the attributes of the adapter:

| Attribute | Details |
|---|---|
| Currently Supported Versions | Itron OpenWay 3.70, 3.90, 4.1, 6.10<br>OpenWay Collection Manager 5.4 |
| Protocol | Proprietary |
| Market(s) | Worldwide |
| Architecture | RF Mesh |

- Itron OpenWay Adapter Processing
- Configuring an Itron OpenWay Head-End System
- Configuring Itron OpenWay Extendable Lookups
- Using the Itron OpenWay Test Harness

### Itron OpenWay Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

#### Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

Measurement and device event data must be requested from the Itron OpenWay head-end system via Scheduled Read commands. See Scheduled Read Commands for more information.

- Scheduled Read Commands
- Initial Measurements


Copyright © 2026, Oracle and/or its affiliates.

- Device Events
- Base Package Business Objects

Scheduled Read Commands

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Itron OpenWay.

Measurement and device event data must be requested from the Itron OpenWay head-end system via Scheduled Read commands. Schedule read commands are used to request meter reads and/or device events for a group of devices for a specified date and time. Scheduled read commands can be created such that recurring requests based on the same command parameters are sent to the head-end system at regular intervals. For example, a command could be configured to request daily (24 hour) scalar meter readings once per day for a specified group of devices. Another command could be configured to request weekly (168 hours) device events once per week for a group of devices.

Scheduled read commands should be created for all devices from which meter readings and/or device events are to be uploaded via Oracle Utilities Smart Grid Gateway. Separate scheduled read commands can be created for specific groups and for specific types of data. For example, for a given group of scalar devices, you might create separate commands to request scalar usage data and to request device events from the same group (both of which can be based on different recurrence patterns). The specifics concerning request specifics and recurrence patterns should be based on the requirements of the implementation.

Attributes used to define a schedule read command include the following:

- **Status:** the status of the activity
- **Schedule Read Type:** the type of scheduled read
- **Schedule Information:** information about the schedule for the read, including:
  - **One Time Read:** indicates if the request is a one-time request. One-time requests have a defined date/time range during which request and recurrences can take place.
  - **One Time Start Date Time:** specifies the start time for one-time requests (applicable only if **One Time Read** is set to “Yes”)
  - **One Time End Date Time:** specifies the end time for one-time requests (applicable only if **One Time Read** is set to “Yes”)
  - **First Daily Measurement Time**: Indicates starting time for measurements for each day. This field, along with the value from **IMD Length**, will be used to determine when interrogation requests should be made. For example, if the **First Daily Measurement Time** is set to 08:00AM and **IMD Length** is set to 12 hours (12:00:00) then there would be two requests made per day:
    - 8:00AM - 8:00PM
    - 8:00PM - 8:00AM
  - **Interrogation Buffer**: This parameter impacts when interrogation requests are sent. It is entered in the format of hours:minutes:seconds. For example, if set to 1 hour (01:00:00) for our above example, the first request of the day for 8:00AM - 8:00PM would be sent to Itron OpenWay at 7:00PM. The buffer time will decrease if the activity is monitored after 7:00PM, so if the monitor process is executed at 7:30PM the buffer


Copyright © 2026, Oracle and/or its affiliates.

time would only be 30 minutes and if the monitor process were not executed until after 8:00PM there would be no buffer time at all.

- **IMD Length**: Defines the number of hours, minutes and seconds of data that should be retrieved for each request. It is entered in the format of hours:minutes:seconds. This number must be divisible into 24 since it will determine how many requests will be made per day (i.e. if set to 8 hours (08:00:00) there will be 3 requests per day).
- **Disable Extended IMD Length Recovery**: By default the schedule read activity will attempt to "catch-up" when interrogation requests have been missed by requesting the entire period from the last interrogation request made. Setting this parameter to "Yes" will turn off this functionality and interrogation requests will only be made in for the exact amount of time defined in the **IMD Length**
- **Range of Recurrence:** indicates how to define request recurrences. Valid values include “Maximum Recurrence”, “No End Date”, and “Recurrence End Date”
  - **Maximum Recurrence**: The activity will make a set number of requests, once the set number has been reached the activity will expire. When selected, **Maximum Recurrences** must also be provided.
  - **No End Date**: The activity will never expire and will continue to make requests until it is manually transitioned to inactive.
  - **Recurrence End Date**: The activity will continue to make requests until a specified date. When selected, **Recurrence End Date Time** must also be provided.

> **Note**
>
> When “No End Data” is selected, the recurrences of the request continue until a user manually changes the status of the command.

- **Maximum Recurrence:** defines the maximum number of recurrences before the command’s status is changed to “Inactive”.
- **Recurrence End Date Time:** The date after which no more recurrences are attempted. After this date and time, the command’s status is changed to “Inactive”.

- **Group:** the group of devices used by requests for the command
  - **Group Type:** indicates the type of device group to use for the command. Valid options include Application Group or Configuration Group.
  - **Application Group/Configuration Group Name:** defines the specific group (based on the group type).

> **Note**
>
> Groups referenced by Scheduled Read commands and the devices that belong to each are defined in Itron OpenWay head-end system. Each group to be used with these commands must also be defined in either the “Itron — Application Group Lookup” or “Asset Specification” extendable lookup. See the Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay Configuration Guide for more information about these extendable lookups.

- **Default Read Parameters:** specifies details concerning the data to be requested. The read parameters define the exact data that will be requested from the meters. Please consult the Itron OpenWay documentation for the specifics of each parameter.


Copyright © 2026, Oracle and/or its affiliates.

  - **Interrogation Window Hours:** the number of hours used to calculate when to submit requests. For example, to submit requests once per day, this should be set to 24. To submit requests once per week (7 days), this should be set to 168. The current date time (or the Latest Request End Date Time) plus this equals interrogation window end date time.

- **Override Read Parameters**: an alternative set of parameters that can be defined. To leverage this group each parameter must be provided, no parameter is allowed to be left blank. This set of parameters will be used when the request end time is equal to the **First Daily Measurement Time**. For example, a schedule read with a **First Daily Measurement Time** of 12:00AM and an **IMD Length** 12 hours (12:00:00) would make the following requests each day:
  - 12:00AM - 12:00PM
  - 12:00PM - 12:00AM

  The **Override Read Parameters** would be used for the 2nd request where the request end time of 12:00AM matches the **First Daily Measurement Time** of 12:00AM. If an extended request is made to "catch up" (i.e. it makes a request that is larger in duration than the configured **IMD Length**), and the request start and request end date time includes the **First Daily Measurement Time** then the **Override Read Parameters** will be used.
- **Outbound Communication Exception Handling Overrides**: parameters used to control retry attempts and expiration for the outbound communication. These parameters take precedence over their outbound communication counterparts
- **Last Read Details:** details concerning the most recent read request sent for the command. This information is used to calculate the time for the next request (based on the **Interrogation Window Hours** parameter).
  - **Latest Request Start Date Time:** the start date and time of the last request
  - **Latest Request End Date Time:** the end date and time of the last request
  - **Recurrence Count:** the number of recurrences of the last request

### About Ad Hoc Requests

While scheduled read commands are typically configured to create recurring requests for meter reads and device events, users can also submit ad hoc requests for a specific date and time range based on the parameters of a currently active scheduled read command. This is useful if meter read data or device events are needed before the next request for a command would be submitted based on the command's recurrence settings.

Ad hoc requests are created as separate one-time request commands, with a start and end time based on parameters supplied by the user when initiating the ad hoc request. The group and read parameters for ad hoc requests are the same as those for the command used to send the ad hoc request.

### Error Handling

If errors occur when processing scheduled read commands, errors are logged as follows:

- If there is a single error it will create a single log entry
- If there is more than one error, the log entry will perform a count of devices and create a single error message stating the number of devices that were in error.

Initial Measurements


Copyright © 2026, Oracle and/or its affiliates.

The usage data received from the AMI head-end system as a file in Itron OpenWay XML format is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D8-ITRONXML-BASE** contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to SGG-D8-ITRONXML-CM.
2. **SGG-D8-ITRONXML-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D8-ITRONXML-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the initial measurement data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElementInterval | Holds the name of inbound web service for the interval IMD seeder. | |
| destinationRootElementScalar | Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval. | |
| destinationRootElementEvent | Holds the name of inbound web service for the device event seeder. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |
| filterUsage | Determines if usage should be filtered. | true<br>false |
| filterRegisterSource | Determines if source register data should be filtered. | true<br>false |

### Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Initial Measurement Data Publishing


Copyright © 2026, Oracle and/or its affiliates.

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D8-ITRONXML-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

**Configuring Initial Measurement Publishing Output**

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Description | Default Value |
|---|---|---|
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |

#### Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **Itron - UOM Code to Standard UOM Mapping** extendable lookup (D8-HeadendUOMLookup) are passed into the system for processing.

#### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

#### Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Device Events

Itron OpenWay can send real-time events such as meter outage alarms to Oracle Utilities Smart Grid Gateway. These events are referred to as exceptions in the Itron architecture, and are sent to the BPEL ExceptionSubscriberService service by Itron OpenWay without requiring


Copyright © 2026, Oracle and/or its affiliates.

an explicit request from Oracle Utilities Smart Grid Gateway. Exceptions are loaded as device events data and provided to the edge applications.

The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D8-EXCEPTION-BASE** contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to the SGG-D8-EXCEPTION-CM project.
2. **SGG-D8-EXCEPTION-CM** allows the customization and simplifies the future upgrades.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the device event data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElement | Holds the name of inbound web service for the device event seeder. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |

### Publishing Events

The Itron OpenWay adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D8-EXCEPTION-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.


Copyright © 2026, Oracle and/or its affiliates.

### Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |


Copyright © 2026, Oracle and/or its affiliates.

### Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **Itron - Device Event Mapping** extendable lookup (D8-DeviceEventMappingLookup) are passed into the system for processing.

### Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D8-EXCEPTION-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.


Copyright © 2026, Oracle and/or its affiliates.

10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

### Base Package Business Objects

The Itron OpenWay adapter base package includes the following initial measurement and device event business objects:

| Business Object Name | Description |
|---|---|
| D8-InterrogateByGroup | Itron - Interrogate By Group<br>Performs a Schedule Read for a group of devices (either Application or Configuration) in the Itron OpenWay API. |
| D8-InterrogateByGroupResult | Itron - Interrogate by Group Result<br>This BO is the inbound asynchronous response to an Interrogate By Group (Schedule Read) outbound communication. |
| D8-InitialLoadIMDInterval | Itron - Initial Load IMD - Interval<br>Used when loading Itron OpenWay interval measurements into the system for the first time. |
| D8-InitialLoadIMDScalar | Itron - Initial Load IMD - Scalar<br>Used when loading Itron OpenWay scalar measurements into the system for the first time. |
| D8-ScheduleRead | Itron - Schedule Read<br>Requests IMD and Event data from the Itron OpenWay Network. This data will later be returned and placed in files to be uploaded. |

## Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes


Copyright © 2026, Oracle and/or its affiliates.

## Communication Flows

The table below lists the communications created for each Itron OpenWay command:

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Remote Connect | Itron - Reconnect Meter | Itron - Reconnect Meter Result | Connect Device Completion Event |
| Remote Disconnect | Itron - Disconnect Meter | Itron - Disconnect Meter Result | Disconnect Device Completion Event |
| Device Commissioning | Itron - Add Meter Definition | | Device Commissioning Completion Event |
| Device Decommissioning | Itron - Deregister Meter | | Device Decommissioning Completion Event |
| On-Demand Read (Scalar) | Itron - Contingency Read (Scalar) | Itron - Contingency Read Result (Scalar) | Create IMD Completion Event |
| On-Demand Read (Interval) | Itron - Contingency Read (Interval) | Itron - Contingency Read Result (Interval) | Create IMD Completion Event |
| Scheduled Read (Scalar) | Itron - Interrogate by Group (Scalar) | Itron - Interrogate by Group Result (Scalar) | Create IMD Completion Event |
| Scheduled Read (Interval) | Itron - Interrogate by Group (Interval) | Itron - Interrogate by Group Result (Interval) | Create IMD Completion Event |
| Device Status Check | Itron - Ping by Endpoints (Status Check) | Itron - Get Ping by Endpoints Result | |

## Device Communication Base Package Business Objects

The Itron OpenWay Adapter base package includes the following communication business objects:

| Business Object Name | Description |
|---|---|
| D8-AddMeterDefinitions | Itron - Add Meter Definition (Commission) |
| D8-DeregisterMeter | Itron - Deregister Meter (Decommission) |
| D8-DetectLoadSideVoltageByMtr | Itron - Detect Load Side Voltage |
| D8-DetLoadSideVoltageMtrRslt | Itron - Detect Load Side Voltage Result |
| D8-DisconnectMeter | Itron - Disconnect Meter (Remote Disconnect) |
| D8-DisconnectMeterResult | Itron - Disconnect Meter Result |
| D8-PingByEndpoints | Itron - Ping By Endpoints (Device Status Check) |
| D8-PingByEndpointsMDResponse | Itron - Multi-Device Ping Response |
| D8-PingByEndpointsMultiDevice | Itron - Ping By Endpoints Multi-Device |
| D8-PingByEndpointsResponse | Itron - Ping By Endpoints Response |
| D8-ReadDisconStateByMtr | Itron - Read Disconnect State |
| D8-ReadDisconStateMtrRslt | Itron - Read Disconnect State Result |
| D8-ReadInterval | Itron - Contingency Read (Interval) |
| D8-ReadResult | Itron - Contingency Read Result |
| D8-ReadScalar | Itron - Contingency Read (Scalar) |


Copyright © 2026, Oracle and/or its affiliates.

| Business Object Name | Description |
|---|---|
| D8-ReconnectMeter | Itron - Reconnect Meter (Remote Connect) |
| D8-ReconnectMeterResult | Itron - Reconnect Meter Result |

### Itron OpenWay Event Data Mapping

The Itron OpenWay event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

| Itron OpenWay Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Transaction ID (from Header record) | External Source Identifier | This is the file name. |
| Device Identifier | External Device Identifier | |
| Event Name | External Event Name | |
| Event Creation Date/Time | Event Date/Time | |
| Device Type | External Device Type | This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form):<br>Meter<br>Collector<br>Router |
| Service Location ID | External Service Location ID | |
| Communication Module Serial Number | External Communication Module Identifier | |
| Event Category ID | External Event Category | |
| Event Severity | External Event Severity | Valid values include (although the element itself is free-form):<br>Alert<br>Information |
| Status Value | External Status Value | This represents additional information that relates to the event itself. |
| Status Date/Time | External Status Date/Time | The date & time at which the additional information referenced above had occurred. |

## External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)


Copyright © 2026, Oracle and/or its affiliates.

- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted.

| Outbound Message Type | Description |
|---|---|
| D8ITRECO | D8 Itron Remote Connect |

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Itron OpenWay includes the following inbound web services:

| Inbound Web Service | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Used for upload of device events.<br>The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event.<br>If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created. |
| D1-DeviceStatusCheck | Device Status Check<br>This service is invoked by the integration layer to instantiate a Device Status Check command. |
| D1-InitialLoadIMD | Used by OSB to instantiate an IMD<br>The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system. |
| D1-RemoteConnect | Remote Connect<br>This service is invoked by the integration layer to instantiate a Remote Connect command. |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service | Description |
|---|---|
| D1-RemoteDisconnect | Remote Disconnect<br>This service is invoked by the integration layer to instantiate a Remote Disconnect command. |
| D8-DetLoadSideVoltageMtrRslt | Itron - Detect Load Side Voltage by Meter Result |
| D8-DisconnectMeterResult | Itron - Disconnect Meter Result Service |
| D8-InterrogateByGroupResult | Itron - Interrogate By Group Result XIA |
| D8-PingByEndpointsMDResponse | Itron - Ping By Endpoints Multi-Device Response XAI Inbound |
| D8-PingByEndpointsResponse | Itron - Ping By End Response inbound web service |
| D8-ReadDisconStateMtrRslt | Itron - Read Disconnect State by Meter Result Service |
| D8-ReadResult | Itron - Contingency Read Result |
| D8-ReconnectMeterResult | Itron - Reconnect Meter Result Service |

### Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway adapter for Itron OpenWay includes the following message senders:

| Message Sender | Description |
|---|---|
| D8-RemoteCon | D8 Remote Connect |

## BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to Itron OpenWay format, invoking process callouts and invoking the remote endpoint to trigger the device events. BPEL modules are divided into “Composites,” or collections of business logic.

**OnDemandRead Composite Process** — The OnDemandRead composite invokes a sequence of web methods that call the head-end system to retrieve meter reading data and send it back to the OUAF layer. In the case of Itron OpenWay, this is accomplished using the ContingencyReadByEndpoints/GetContingencyReadByEndpointsResult services defined in the Data service WSDL.

**ConnectDisconnect Composite Process** — This composite is responsible for triggering the Connect and Disconnect events on the head-end system. A second, asynchronous reply sends the results back into the OUAF layer when the head-end system signals a change in the status of the device. The Itron OpenWay Adapter uses the ReconnectMeter/GetReconnectMeterResult web service pair to connect, and the DisconnectMeter/GetDisconnectMeterResult web service pair to disconnect. Each of these services is defined in the Control service WSDL.

**CommissionDecommission Composite Process** — Commissioning and Decommissioning of devices, sometimes referred to as “Provisioning,” is handled by the CommissionDecommission composite. Commissioning invokes the AddMeterDefinitions web


Copyright © 2026, Oracle and/or its affiliates.

service and Decommissioning uses the DeregisterMeters web service. Both are defined in the Provisioning service.

**DeviceStatusCheck Composite** — This composite uses the PingByEndpoints and GetPingByEndpointsResult web services to check the health of a device. These Itron OpenWay services are defined in the Control.Diagnostic service.

**Common Composite** — The Common composite contains three main classes of operations: Proxies, ProcessCallouts, and utility functions. Proxies are usually simple mediators that forward a web service call to a preset endpoint. In this Adapter they have two additional roles. First, they interrogate a composite property and determine whether the supported version of head-end software is 3.70 or 3.90. Some of the proxies will also append the callback URL for the StatusChanged service. Proxies are convenient because they allow head-end URLs and security to be set in a single composite. In this case, they also offer a common location to make these checks. ProcessCallouts are points of customization which allow users to modify data and/or initiate some external business process. Utility functions serve as a central location for business logic needed by multiple composites.

### Web Services

The following web services are all defined in the Itron OpenWay head-end system:

- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** AddMeterDefinitions
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/CommissionDecommission/CommissionDecommissionService
- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** DeregisterMeters
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/CommissionDecommission/CommissionDecommissionService
- **ConnectDisconnectService**
  - **BPEL Process:** ConnectDisconnect
  - **Operation:** ReconnectMeter
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/ConnectDisconnect/ConnectDisconnectService
- **ConnectDisconnectService**
  - **BPEL Process:** ConnectDisconnect
  - **Operation:** DisconnectMeter
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/ConnectDisconnect/ConnectDisconnectService
- **DeviceStatusCheckService**
  - **BPEL Process:** DeviceStatusCheck
  - **Operation:** DeviceStatusCheck
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/DeviceStatusCheck/DeviceStatusCheckService


Copyright © 2026, Oracle and/or its affiliates.

- **OnDemandReadService**
  - **BPEL Process:** OnDemandRead
  - **Operation:** ContingencyReadByEndpoints
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/<EM_PARTITION>/OnDemandRead/OnDemandReadService

### Itron OpenWay Web Services

The following table describes Itron OpenWay web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

| Smart Grid Gateway Command | AMI Adapter Business Objects | Itron OpenWay Web Services | Itron OpenWay Operations |
|---|---|---|---|
| Device Commissioning | D8-AddMeterDefinitions | ProvisioningService | AddMeterDefinitions |
| Device Decommissioning | D8-DeregisterMeter | ProvisioningService | DeregisterMeters |
| Remote Connect | D8-ReconnectMeter | ControlService | ReconnectMeter |
| | D8-DetectLoadSideVoltageByMtr | DiagnosticService | DetectLoadSideVoltageByMeter |
| | D8-ReadDisconStateByMtr | | ReadDisconnectStateByMeters |
| Remote Disconnect | D8-DisconnectMeter | ControlService | DisconnectMeter |
| Device Status Check | D8-PingByEndpoints | ControlService | PingByEndpoints |
| | | | GetPingByEndpointsResult |
| On-Demand Read | D8-ReadInterval | DataService | ContingencyReadByEndpoints |
| | D8-ReadScalar | | GetContingencyReadByEndpointsResult |
| | D1-InitialLoadIMD | | InterrogateByGroup |
| | D1-DeviceEventSeeder | | GetInterrogateByGroupResult |
| | | DataSubscriberService (Async Response) | DataArrived |
| Event Loading | D1-DeviceEventSeeder | ExceptionSubscriberService | ExceptionsArrived |

## Configuring an Itron OpenWay Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay to communicate with the Itron OpenWay application.

- Master Configurations
- Inbound Web Services
- Message Senders
- Outbound Message Types


Copyright © 2026, Oracle and/or its affiliates.

- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

## Master Configurations

Master Configurations are sources of global parameter records used by a system implementation. This section describes the master configuration that is specific to Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about other master configurations used by Oracle Utilities Smart Grid Gateway.

### Itron Version Master Configuration

This master configuration specifies the version of the Itron head-end system with which the system is communicating. The configuration options are Itron OpenWay 3.70, Itron OpenWay 3.90, and Itron OpenWay 6.10. Only one version can be specified at a time.

## Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the Itron OpenWay application in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

| Inbound Web Service Name | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Device Event Seeder |
| D1-DeviceStatusCheck | Device Status Check |
| D1-InitialLoadIMD | Used by OSB to instantiate an IMD |
| D1-PayloadErrorNotif | Payload Error Notification |
| D1-PayloadStatistics | Payload Statistics |
| D1-PayloadSummary | Payload Summary |
| D8-DetLoadSideVoltageMtrRslt | Itron - Detect Load Side Voltage by Meter Result |
| D8-DisconnectMeterResult | Disconnect Meter Result Service |
| D8-InterrogateByGroupResult | Itron - Interrogate By Group Result |
| D8-PingByEndpointsMDResponse | Itron - Ping By Endpoints Multi-Device Response XAI Inbound |
| D8-PingByEndpointsResponse | Itron - Ping By Endpoints Response inbound web service |
| D8-ReadDisconStateMtrRslt | Itron - Read Disconnect State by Meter Result Service |
| D8-ReadResult | Itron - Contingency Read Result |
| D8-ReconnectMeterResult | Itron - Reconnect Meter Result Service |


Copyright © 2026, Oracle and/or its affiliates.

Note: The following apply to all of the above inbound web services:

**Message Options**

- **Trace**: No
- **Debug**: No
- **Active**: Yes

**Operations**

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

## Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests. A message sender should be configured for each command.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

| Message Sender | Description |
|---|---|
| D8-ITRON_IBG | Itron Interrogate By Group |
| D8-COMM | Itron Commission |
| D8-DCOMM | Itron Decommission |
| D8-DLSV | Itron Detect Load Side Voltage |
| D8-IPBE | Itron Ping By Endpoints |
| D8-RCONN | Itron Remote Connect |
| D8-RDCONN | Itron Remote Disconnect |
| D8-RDSS | Itron Read Disconnect State |

**Note:** The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/Itron/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST


Copyright © 2026, Oracle and/or its affiliates.

- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/Itron/**<SERVICE>**
- where:
  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
  - **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

**How to Use Enterprise Manager to Locate the URL for the Message Sender**

Follow this procedure to find the correct URL for the command associated with a message sender:

1. Open Enterprise Manager and use the navigation pane to open the dashboard of the service used by the message sender:
2. The top bar of the dashboard contains several buttons and icons. One of these is a “world” icon with a puzzle piece over it. Click this icon to display a list of the WSDLs and endpoint URIs for the service:
3. Click the service’s WSDL URL link to see the WSDL in the browser, or right click and save it to your machine.

## Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

| Outbound Message Type | Description |
|---|---|
| D8-CONREAD | Itron Contingency Read By Endpoints |
| D8-INTRGRP | Itron Interrogate By Group |
| D8-COMM | Itron Commission |
| D8-DCOMM | Itron Decommission |
| D8-DSC | Itron Device Status Check |
| D8-DSCMD | Itron Device Status Check Multi Device |
| D8-DLSV | Itron Detect Load Side Voltage |
| D8-RCONN | Itron Remote Connect |
| D8-RDCONN | Itron Remote Disconnect |
| D8-RDSS | Itron Read Disconnect State |

Note: The following apply to all of the above outbound message types:


Copyright © 2026, Oracle and/or its affiliates.

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

## External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the Itron OpenWay head end must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

**External System - Itron OpenWay**:

- **External System**: Itron OpenWay
- **Description**: Itron OpenWay
- **Outbound Message Types**:

| Outbound Message Type | Message Sender |
|---|---|
| D8-INTRGRP | D8-ITRON_IBG |
| D8-COMM | D8-COMM |
| D8-DCOMM | D8-DCOMM |
| D8-DSC | Message sender associated with the Device Status Check Outbound Message Type |
| D8-DSCMD | Message sender associated with the Device Status Check Multi Device Outbound Message Type |
| D8-DLSV | D8-DLSV |
| D8-RCONN | D8-RCONN |
| D8-RDCONN | D8-RDCONN |
| D8-RDSS | D8-RDSS |

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D8-Request xsl
- **Response XSL**: D8-Response.xsl

## Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the Itron OpenWay head end must be present in your system. If this are not present in your configuration, add it. Refer to the Oracle Utilities Service


Copyright © 2026, Oracle and/or its affiliates.

and Measurement Data Foundation documentation for more information about creating service providers.

**Service Provider - Itron OpenWay**:

- **Service Provider**: Itron OpenWay
- **Description**: Itron OpenWay
- **External Reference ID**: Itron OpenWay
- **External System**: Itron OpenWay
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **Pre-Commissioning Device ID Type**:
- **AMI Measuring Component ID Type**: Channel ID

## Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the Itron OpenWay service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

### Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

### Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

### UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

### Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the Itron OpenWay service provider, based on the requirements of each implementation using the “How to Create


Copyright © 2026, Oracle and/or its affiliates.

OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

| Command | Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|---|
| Device Commission | Device Registration | D8-AddMeterDefinitions | Commission |
| Device Decommission | Device Removal | D8-DeregisterMeter | Decommission |
| Device Status Check | Device Status Check | D8-PingByEndpoints | Device Status Check |
| Load Check* | Load Check | D8-DetectLoadSideVoltageByMtr | Detect Load Side Voltage by Meter |
| Multi-Devie Status Check | Multi-Devie Status Check | D8-PingByEndpointsMultiDevice | Read Disconnect State By Meters |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | D8-ReadScalar | On Demand Read — Scalar |
| Remote Connect | Remote Connect | D8-ReconnectMeter | Connect |
| Remote Disconnect | Remote Disconnect | D8-DisconnectMeter | Disconnect |

* The Load Check processing method can be used to ensure that it is safe to connect a meter. The load check processing method is executed during the Connection Ready state of the Remote Connect activity and can be configured to detect the possibility of a load side voltage at the meter, or to check the disconnect meter switch on the meter.

If the Read Disconnect State business object is used as the processing method, you should configure the disconnect switch state values that indicate the state of the switch. These values are defined on the D8-DisconnectSwitchStateLookup extendable lookup, and are then added in the Disconnect Switch State Error Values section of the Itron - Read Disconnect State Result Inbound communication type. See Itron OpenWay Disconnect Switch State Lookup for more information.

## Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Itron OpenWay endpoint override DVM (D8–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Itron OpenWay adapter and the command used with each:


Copyright © 2026, Oracle and/or its affiliates.

| DVM Key | Commands |
|---|---|
| DataService390 | Scheduled Read (Scalar)<br>Scheduled Read (Interval)<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| ProvisioningService370<br>ProvisioningService390 | Device Commissioning<br>Device Decommissioning |
| ControlService370 | Device Status Check (if using v3.70)<br>Remote Connect (if using v3.70)<br>Remote Disconnect (if using v3.70) |
| ControlService390 | Remote Connect (if using v3.90)<br>Remote Disconnect (if using v3.90) |
| DiagnosticService390 | Device Status Check (if using v3.90) |
| ProcessCallout | User Exit Functions |

> **Note**
>
> The numbers in the keys above (370 and 390) designate the version of the Itron OpenWay head-end system.
>
> Only a single version of each key should be defined, based on the version of the Itron OpenWay head-end system

To define an override Endpoint URI for the Itron OpenWay adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D8–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon (“+”) to add a new key.

   The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.

6. Click **OK**.


Copyright © 2026, Oracle and/or its affiliates.

The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

   Click **Discard** to discard your changes.

   Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

## Configuring Itron OpenWay Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Itron OpenWay.

This section outlines some of the extendable lookups that must be configured for use with the Itron OpenWay adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

### Application Group Lookup

The Itron OpenWay head-end system allows users to define addressable groups of meters based on criteria such as a geographic identifier or zip code. These application groups can be represented in Oracle Utilities Smart Grid Gateway by using the Itron OpenWay Application Group Lookup. The groups defined for this lookup are used with Scheduled Read commands. Each value defined for the Itron OpenWay Application Group extendable lookup should include the following:

- **Application Group:** The application group name as defined in the Itron OpenWay head-end system.
- **Description:** A description of the application group.

### Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

### Disconnect Switch State Lookup

Some utilities may require verification of the load on a meter to ensure that it is safe to connect the meter. One method of doing this is to check the disconnect switch on the meter to see if it is connected or armed. To use this method, the Itron Openway Disconnect Switch State


Copyright © 2026, Oracle and/or its affiliates.

extendable lookup should be configured to list the possible states of the meter disconnect switch. Each value defined for the Itron OpenWay Interval Disconnect Switch State extendable lookup should include the following:

- **Switch State**: The Itron OpenWay disconnect switch state code
- **Description**: A description of the disconnect switch state code.

### UOM Code to Standard UOM Mapping

Usage received from a utility's head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-End Unit of Measure**: The unit of measure code used by the Itron OpenWay application
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Unit of Measure**: The unit of measure defined in the system.
- **Time of Use**: An optional time of use period, defined in the system, used to further distinguish the unit of measure.
- **Service Quantity Identifier**: An optional service quantity identifier, defined in the system, used to further distinguish the unit of measure.
- **Unit of Measure Magnitude Conversion**:
  - **Magnitude Multiplier**: An optional; multiplier used to convert raw values received from the head-end system to values appropriate for use with the system. For example, if a reading or interval data is received in Wh, a multipler of ".001" would convert Wh to kWh. If not provided or left blank, no conversion is performed.

### Interval Status Code to Condition Mapping

Interval usage received from the Itron OpenWay application can include Itron OpenWay interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Itron OpenWay Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Itron OpenWay interval status codes to standard status codes when receiving usage from the Itron OpenWay application. Each value defined for the Itron OpenWay Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The Itron OpenWay interval status code
- **Description**: A description of the interval status code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

### Other Extendable Lookups

| Business Object Name | Description |
|---|---|
| D8-DvcStatusMappingLookup | Itron - Device Status Lookup |
| D8-FailureReasonLookup | Itron - Failure Reason Lookup |


Copyright © 2026, Oracle and/or its affiliates.

| Business Object Name | Description |
|---|---|
| D8-ItronVersions | Itron - Versions Lookup |
| D8-JobStatusLookup | Itron - Job Status |
| D8-ResultLookup | Itron - Result Lookup |
| D8-RptSelfReadOptionLookup | Itron - Report Self Read Option Lookup |

### Using the Itron OpenWay Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay includes a test harness that can be configured to simulate a general head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Test Harnesses for more information.

## Landis+Gyr

The Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr supports communication with the Landis+Gyr Gridstream Command Center, including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, and on-demand read. The following table describes the attributes of the adapter:

| Attribute | Details |
|---|---|
| Currently Supported Version | Gridstream Command Center 7.5, 8.1, 8.4, 8.5, 9.0.4 |
| Smart Meter Command Format | MultiSpeak 3.0<br>CIM 1.ov2<br>CIM 2.0 |
| Bulk Usage/Event Data Format | California Metering Exchange Protocol (CMEP) |
| Market(s) | North America, portions of Asia Pacific, Sweden, Australia, New Zealand, and Latin America. |
| Commodities Supported | Electricity, Gas, Water |
| Architecture | PLC and RF |

- Landis+Gyr Adapter Processing
- Configuring a Landis+Gyr Head-End System
- Configuring Landis+Gyr Extendable Lookups
- Using the Landis+Gyr Test Harness
- Landis+Gyr Interval Data Mapping

### Landis+Gyr Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.


Copyright © 2026, Oracle and/or its affiliates.

- Initial Measurement Data and Device Event Loading
- Device Communication

## Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements
- Device Events
- Base Package Business Objects

### Initial Measurements

The usage data exported from the AMI head-end system as a file in Landis+Gyr format is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D3-USAGE-BASE** - contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This can be upgraded without affecting the customization and environment settings added to SGG-D3-USAGE-CM.
2. **SGG-D3-USAGE-CM** allows for customization and simplifies future upgrades.

When importing non-interval usage data, separate initial measurements can be created for difference measurement types. For instance, if the data includes Power Factor or Volt data, separate initial measurements are created for each of these. See Non-Interval 'Plain' XML to IMD Mapping for more information about specific units of measure that trigger the creation of separate initial measurements.

The runtime configuration settings for the SGG-D3-USAGE-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRawIMD | Determines if the initial measurement data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElementInterval | Holds the name of inbound web service for the interval IMD seeder. | |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| destinationRootElementScalar | Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval. | |
| modifyResultXMLInput | Specifies the name of an XQuery document (without the "xq" extension) used to map additional fields from the "plain" XML format to the result XML format sent as initial measurement data. See Mapping Additional Fields for more information. | |
| dateTimeInUTC | Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. | true<br>false |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterUsage | Determines if usage should be filtered. | true<br>false |

### Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D3-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

### Configuring Initial Measurement Publishing Output


Copyright © 2026, Oracle and/or its affiliates.

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |


Copyright © 2026, Oracle and/or its affiliates.

### Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **Landis+Gyr UOM Code to Standard UOM Mapping** extendable lookup (D3-HeadendUOMLookup) are passed into the system for processing.

> **Note**
>
> Filtering of scalar initial measurement data is not supported in the Landis+Gyr adapter.

### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

### Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

## Device Events

The device event data exported from the head-end system as a file in Landis+Gyr format is loaded into Oracle Utilities as a Device Event. One of your configuration tasks is to customize the device events processing. The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D3-EVENT-BASE** containing components responsible for "actual" processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D3-EVENT-CM project.
2. **SGG-D3-EVENT-CM** allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D3-EVENT-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element. The following table describes the elements included in the EnvironmentSettings.xq file:


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the device event data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElement | Holds the name of inbound web service for the device event seeder. | |
| modifyResultXMLInput | Specifies the name of an XQuery document (without the "xq" extension) used to map additional fields from the "plain" XML format to the result XML format sent as device event data. See Mapping Additional Fields for more information. | |
| dateTimeInUTC | Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. | true<br>false |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |

### Publishing Events

SGG can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D3-EVENT-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.


Copyright © 2026, Oracle and/or its affiliates.

- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

### Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |


Copyright © 2026, Oracle and/or its affiliates.

### Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **Landis+Gyr Device Event Mapping** extendable lookup (D3-DeviceEventMappingLookup) are passed into the system for processing.

### Subscribing to Real-Time Device Events

The Landis+Gyr Command Center provides the ability to "subscribe" to device events from within their online interface. This is done by creating a subscriber in Command Center with an associated callback URL as well as a list of events types that subscriber is interested to receive. When an event that is subscribed to occurs it is sent to the callback URL in real time.

Within L+G Events are configured with one of three "alarm" settings. These settings determine how often the events will be sent to subscribers:

1. Alarm: immediately delivered from the meter
2. Advisory: sent based upon a delivery schedule
3. Log Only: sent only upon request (not applicable for our implementation real time event processing)

Command Center will communicate the events using a CIM format that describes the message as a noun/verb combination. The details of the event itself will be contained within a "payload" element of the standard structure. The payload will be formatted using the EndDeviceEvent message structure. This message identifies device events using a CIM 4-part category number. These numbers are four period separated numbers that will describe the type of device and the event. For example: 3.33.1.257 is for "Tamper attempt suspected".

- Segment 1: End Device event domain code (e.g. 3. meter/10. collector/11. router/12. HAN device)
- Segment 2: End Device Event Domain Part Codes (e.g. 1. Access/2. Battery)
- Segment 3: End Device Event Type Codes (e.g. 1. Alarm/2. Alarm Mgt)
- Segment 4: End Device Event Index (e.g. 1. Abort/2. Access Attempt)

Refer to the Landis+Gyr documentation for details about the CIM Category Numbers. CIM Category Numbers must be mapped to standard device event names using the Landis+Gyr Device Event Mapping extendable lookup.

SGG receives these messages through a BPEL composite that saves the incoming request as a file to be picked up by OSB.

The **AMIEventSubscriber** composite is responsible for receiving the event messages based on subscriptions defined in the L+G Command Center. The callback URL configured for the subscription in the Command Center should point to this BPEL composite.

The following OSB projects parse individual device events from the message and perform the validation and mapping of the information to the Device Event Seeder Format.


Copyright © 2026, Oracle and/or its affiliates.

1. **SGG-D3-CIM-EVENT-BASE** contains components responsible for "actual" processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D3-CIM-EVENT-CM project.
2. **SGG-D3-CIM-EVENT-CM** allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D3-CIM-EVENT-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the device event data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElement | Holds the name of inbound web service for the device event seeder. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |

Processing statistics are gathered for any real time events that are received (even if there is just one event in the message) in the same manner as device events received via the flat-file interface.

### Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)


Copyright © 2026, Oracle and/or its affiliates.

- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D3-CIM-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** colulmn.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects

The Landis+Gyr adapter base package includes the following initial measurement business objects:

| Business Object Name | Description |
|---|---|
| D3-InitialLoadIMDInterval | Landis+Gyr Initial Load IMD - Interval<br>Used when loading Landis+Gyr interval measurements into the system for the first time. |
| D3-InitialLoadIMDScalar | Landis+Gyr Initial Load IMD - Scalar |

## Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows


Copyright © 2026, Oracle and/or its affiliates.

- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes

Communication Flows

The table below lists the communications created for each Landis+Gyr command:

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Remote Connect (This command has sub-commands) | Initiate MR by Mtr Num<br>Initiate Connect Disconnect | Reading Changed Notification<br>Connect Disconnect State Changed Notification | Connect Device<br>Create IMD Completion Event |
| Remote Disconnect (This command has sub-commands) | Initiate Connect Disconnect<br>Initiate MR by Mtr Num | Connect Disconnect State Changed Notification<br>Reading Changed Notification | Disconnect Device<br>Create IMD Completion Event |
| Device Commissioning (Registration) | L+G Add Meter to Inventory | | Device Commissioning |
| Device Commissioning (Installation) | L+G Meter Exchange Notification | | Device Commissioning |
| Device Decommissioning | Meter Remove Notification | | Device Decommissioning |
| Device Deregistration | L+G Meter Retire Notification | | Device Deregistration |
| On-Demand Read (Scalar) | Initiate MR by Mtr Num | Reading Changed Notification | Create IMD Completion Event |
| On-Demand Read (Scalar) - CIM | CIM Meter On Demand Read (Scalar) | CIM Meter On Demand Read Response | Create IMD Completion Event |
| On-Demand Read (Interval) | Initiate MR by Mtr Num | Reading Changed Notification | Create IMD Completion Event |
| On-Demand Read (Interval) - CIM | CIM Meter On Demand Read (Interval) | CIM Meter On Demand Read Response | Create IMD Completion Event |
| Device Status Check | CIM Ping | CIM Ping Response | |
| Demand Reset | Schedule Demand Reset (Multispeak) | Schedule Demand Reset Response (Multispeak) | Create IMD Completion Event |

### Device Registration Commission Commands

Landis+Gyr device commission commands can be used to “register” the device and notify the L+G head-end system that meters have been added to inventory. Device commission commands of this type have the “Registration-Only Mode” flag set to “Yes”. An Enter algorithm on the “Commission Ready” state evaluates the Registration-Only Mode of the command and if set to “Yes”, the command skips the default “Waiting for Measurement” state and is transitioned to the “Execute Completion Event” state, and an activity log entry is created.


Copyright © 2026, Oracle and/or its affiliates.

Only the device registration request is sent to the head-end system for device commission commands of this type.

Device registration commands are typically created when new devices are added to inventory in an asset management system such as Oracle Utilities Operational Device Management.

### Device Installation Commission Commands

Landis+Gyr device commission commands can be used to notify the L+G head-end system that meters have been installed or exchanged. An Enter algorithm on the “Commission Ready” state evaluates the “Is Installation Check Unnecessary” flag of the command and if set to “False”, the algorithm creates an “L+G Meter Exchange Notification” outbound communication and sends an installation notification to the head-end system.

### Device Deregistration Commands

The Landis+Gyr adapter supports the Device Deregistration command (based on the D1-DeviceDeregistration business object). This command sends a communication that deregisters the device in the head-end system, and is most often used when retiring a device. The specific message sent is defined for the Device Deregistration processing role for the L+G head-end system service provider.

Device deegistration commands are typically created when devices are retired in an asset management system such as Oracle Utilities Operational Device Management.

### Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CCOMPEV (Create IMD Completion Event ) algorithm on the Create Completion Events status of the D3-ReadingChgNotification (Reading Changed Notification - SOA based) business object.

The D3-VALDVCFIL (Check if Device has any Filtering errors) algorithm on the Validate status of the D3-ReadingChgNotification (Reading Changed Notification - SOA based) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

### Demand Reset Commands

For on-demand read commands associated with Demand Reset commands, filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-SDRCCE (Create Schedule Demand Reset Completion Event) algorithm on the Create Completion Events status of the D3-ScheduleDemandResetResponse (Schedule Demand Response(Multispeak)) business object.

The D3-VALSDRCCE (Check if Device has any Filtering errors - Schedule Demand Reset) algorithm on the Validate status of the D3-ScheduleDemandResetResponse (Schedule Demand Response(Multispeak)) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-SDRCCE algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

### CIM Commands


Copyright © 2026, Oracle and/or its affiliates.

For CIM-based on-demand read commands, filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CCMPEVCIM (Create CIM Completion Event) algorithm on the Create Completion Events status of the D3-CIMMeterReadingsResponse (CIM Meter On Demand Read Response SOA Based) business object.

The D3-VALCEVCIM (Check if Device has any Filtering errors CIM Completion Event) algorithm on the Validate status of the D3-CIMMeterReadingsResponse (CIM Meter On Demand Read Response SOA Based) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-CCMPEVCIM algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

Device Communication Base Package Business Objects

The Landis+Gyr Adapter base package includes the following communication business objects:

| Business Object Name | Description | Protocol |
|---|---|---|
| D3-AddMeterToInventoryMultiSp | L+G Add Meter to Inventory (MultiSpeak) | MultiSpeak |
| D3-CIMGetLPData | CIM Meter On Demand Read (Interval) | CIM |
| D3-CIMMeterOnDemandRead | CIM Meter On Demand Read (Scalar) | CIM |
| D3-CIMMeterReadingResponse | CIM Meter On Demand Read Response | CIM |
| D3-CIMPing | CIM Ping | CIM |
| D3-CIMPingResponse | CIM Ping Response | CIM |
| D3-ConnectDisconStateChgNtf | Connect Disconnect State Changed Notification | MultiSpeak |
| D3-InitiateConnectDisconnect | Initiate Connect Disconnect | MultiSpeak |
| D3-InitiateMRByMtrNbr | Initiate Meter Read By Meter (MultiSpeak) | MultiSpeak |
| D3-MeterAddNotificationMultiSp | Meter Add Notification (MultiSpeak) | MultiSpeak |
| D3-MeterExNotificationMultiSp | L+G Meter Exchange Notification | MultiSpeak |
| D3-MeterRetireNotification | L+G Meter Retire Notification | MultiSpeak |
| D3-MtrRmvNotifMultiSpeak | Meter Remove Notification (MultiSpeak) | MultiSpeak |
| D3-ReadingChgNotification | Reading Changed Notification | MultiSpeak |
| D3-ScheduleDemandReset | Schedule Demand Reset (Multispeak) | MultiSpeak |
| D3-ScheduleDemandResetResponse | Schedule Demand Reset Response (Multispeak) | MultiSpeak |

### Landis+Gyr Event Data Mapping

The Landis+Gyr event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

| Landis+Gyr Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Transaction ID (from Header record) | External Source Identifier | This is the file name. |
| Device Identifier | External Device Identifier | |


Copyright © 2026, Oracle and/or its affiliates.

| Landis+Gyr Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Event Name | External Event Name | |
| Event Creation Date/Time | Event Date/Time | |
| Device Type | External Device Type | This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form):<br>Meter<br>Collector<br>Router |
| Service Location ID | External Service Location ID | |
| Communication Module Serial Number | External Communication Module Identifier | |
| Event Category ID | External Event Category | |
| Event Severity | External Event Severity | Valid values include (although the element itself is free-form):<br>Alert<br>Information |
| Status Value | External Status Value | This represents additional information that relates to the event itself. |
| Status Date/Time | External Status Date/Time | The date & time at which the additional information referenced above had occurred. |

### External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

### Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted. These notifications are based on the following outbound message types.

| Outbound Message Type | Description |
|---|---|
| D3-ADDMTRINV | Add Meter to Inventory |
| D3-COMMS | Commission Device |
| D3-CONNECT | Connect Device |
| D3-DECOMMS | Decommission |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Type | Description |
|---|---|
| D3-DEMRESET | Demand Reset |
| D3-DERDEV | Deregister Device |
| D3-DISCONNEC | Disconnect Device |
| D3-DVCSTSCHK | Device Status Check |
| D3-INITMRN | Initiate Meter Read by Meter Number |
| D3-INITMTR | Initiate Meter Read by Meter Number |
| D3-MTRADDNOT | Meter Add Notification Outbound Message Type |
| D3-MTREX | Meter Exchange Notification OB MSG |
| D3-MTRRMV | Meter Remove Notification |

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Landis+Gyr includes the following inbound web services:

| Inbound Web Service | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Used for upload of device events.<br>The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event.<br>If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created. |
| D1-InitialLoadIMD | Used for initial measurement upload.<br>The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system. |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service | Description |
|---|---|
| D3-CIMMeterReadingsResponse | CIM Meter On Demand Read Response<br>Retrieve response from CIM On Demand Read command |
| D3-CIMPingResponse | CIM Ping Response<br>Retrieve response from CIM Device Status Check command |
| D3-ConDisconStChgNotification | Initiate Connect Disconnect response.<br>Retrieve response from the Initiate Connect Disconnect command. |
| D3-ReadingChangedNotification | Reading Changed Notification<br>Notification that a Landis+Gyr device reading has changed. |
| D3-ScheduleDemandResetResponse | Scheduled Demand Reset Response<br>Retrieve response from Demand Reset command |

### Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway adapter for Landis+Gyr includes the following message senders:

| Message Sender | Description |
|---|---|
| D3-Comms | Commission Device |
| D3-Connect | Connect Device |
| D3-Decomm | Decommission Device |
| D3-Decomms | Decommissioning Sender |
| D3-DemReset | Demand Reset |
| D3-DerDevice | Deregister Device |
| D3-Disconnec | Disconnect Device |
| D3-InitMTR | Initiate Meter Read by Meter Number Outbound Message |
| D3-MTREXMS | Meter Exchange Notification Message Sender |
| D3-RTSender | Real Time Sender |
| D3-RTSnd | Real-time Sender (Landis+Gyr) |
| D3-SDemReset | SG Demand Reset |

## BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to MultiSpeak 3.0 format, invoking process callouts and invoking the remote endpoint to trigger the device events.

**OnDemandRead Composite Process**: Invokes the remote endpoint to trigger the on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.


Copyright © 2026, Oracle and/or its affiliates.

**ConnectDisconnect Composite Process**: Invokes the remote endpoint to trigger the connect/disconnect event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

**CommissionDecommission Composite Process**: Invokes the remote endpoint to trigger the commission or decommission event. After the synchronous call completes, one of the following second business callout services is invoked to determine if the related “received” or “completed” callout should be executed:

- isExecutingCommissionReceivedCallout
- isExecutingCommissionCompletedCallout
- isExecutingDecommissionReceivedCallout
- isExecutingDecommissionCompletedCallout
- isExecutingAddMeterToInventoryReceivedCallout
- isExecutingAddMeterToInventoryCompletedCallout
- isExecutingMeterExchangeNotificationReceivedCallout
- isExecutingMeterExchangeNotificationCompletedCallout

**CIMOnDemandRead Composite Process**: Invokes the remote endpoint to trigger the CIM on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

**CIMDeviceStatusCheck Composite Process**: This process is similar to CIM OndemandRead BPEL process. SGG uses the result of On Demand Read command to identify the status of the meter. If reads are successfully returned, then meter is running healthy otherwise it is considered as meter inactive/dead.

**DemandReset Composite Process**: Invokes the remote endpoint to trigger the demand reset event. An asynchronous reply responds to the OUAF layer when the reading arrives.

**LGProcessCallout Composite**: This business callout provides a point at which customers and implementers can incorporate custom business logic and transformations. This composite includes the WSDLs and processing logic for all of the MultiSpeak processes. The default implementation of each method is a direct return of the input.

### Web Services

These web services are all defined in the Landis+Gyr head end system. The WSDLs were added to a Meta Data Storage (MDS) layer in OUAF and all references to the WSDL point to this MDS location.


Copyright © 2026, Oracle and/or its affiliates.

| Web Service | Related BPEL Process | Description |
|---|---|---|
| MR_CB | OnDemandRead<br>CommissionDecommission<br>DemandReset | This web service is defined by the Landis+Gyr head end system's implementation of MR_Server.<br>The WSDL defines the interface for requesting a meter reading from the head end system.<br>The actual definition can be obtained from L&G or downloaded from multispeak.org. Build 3.0aa is appropriate if obtained from MultiSpeak.<br>Default endpoint must be changed in configuration: http://demo.turtletech.com/Multispeak/webapi/MR_CB.asmx |
| CD_CB | ConnectDisconnect | This web service is defined by the Landis+Gyr implementation of CB_CD.<br>The WSDL defines the interface for requesting a meter's connection or disconnection on the head end system.<br>This web service defines the interface for reporting a connection or disconnection by the head end system.<br>This web service is only invoked by the head end system; not OUAF.<br>Only the CDStateChangedNotification web method is implemented in the composite.<br>Default endpoint must be changed in configuration: http://demo.turtletech.com/Multispeak/webapi/CD_CB.asmx |
| CIMService | CIMOnDemandRead<br>CIMDeviceStatusCheck | This web service is defined by the L+G head end's implementation of AMIRequest Server.<br>The WSDL defines the interface for requesting a meter reading from the head end system.<br>The actual definition should be obtained from L&G or downloaded from L&G SDK for CIM 2.0. |
| LGProcessCallout | OnDemandRead<br>ConnectDisconnect<br>CommissionDecommission | Imported from **LGProcessCallout Composite**<br>Default endpoint must be changed in configuration: http://127.0.0.1:8000/soa-infra/services/default/LGProcessCallout/LGProcessCallout |

### Landis+Gyr Command Center Web Services

The following table describes the Land+Gyr Command Center web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:


Copyright © 2026, Oracle and/or its affiliates.

| Smart Grid Gateway Command | AMI Adapter Business Objects | Landis+Gyr Web Services | Landis+Gyr Operations |
|---|---|---|---|
| Device Commissioning | D3-MeterAddNotificationMultiSp | MR | MeterAddNotification |
| Device Decommissioning | D3-MtrRmvNotifMultiSpeak | MR | MeterRemoveNotification |
| Remote Connect/ Remote Disconnect | D3-InitiateConnectDisconnect | CD | InitiateConnectDisconnect |
| On-Demand Read | D3-InitiateMRByMtrNbr | MR | InitiateMeterReadByMeterNumber |
| On-Demand Read (CIM) | D3-CIMGetLPData<br>D3-CIMMeterOnDemandRead | CIMService | ScheduleDemandRead |
| Demand Reset | D3-ScheduleDemandReset | MR | CIM |

## Configuring a Landis+Gyr Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr to communicate with the Landis+Gyr Command Center software.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the L+G Command Center in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

| Inbound Web Service Name | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Device Event Seeder |
| D1-InitialLoadIMD | IMD Seeder |
| D1-PayloadErrorNotif | Payload Error Notification |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service Name | Description |
|---|---|
| D1-PayloadStatistics | Payload Statistics |
| D1-PayloadSummary | Payload Summary |
| D3-ConDisconStChgNotification | Initiate Connect Disconnect Response |
| D3-CIMMeterReadingsResponse* | CIM Meter On Demand Read Response |
| D3-CIMPingResponse | CIM Ping Respose |
| D3-ReadingChangedNotification* | Reading Changed Notification |
| D3-ScheduleDemandResetResponse | Scheduled Demand Reset Response |

*The Landis+Gyr adapter supports both MultiSpeak and CIM On Demand Read commands. You only need to configure the inbound service for the protocol you wish to use.

**Note**: The following apply to all of the above inbound web services:

**Message Options**

- **Trace**: No
- **Debug**: No
- **Active**: Yes

**Operations**

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

## Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

| Message Sender | Description |
|---|---|
| D3-Comms | Commission Device |
| D3-Connect | Connect Device |
| D3-Decomm | Decommission Device |
| D3-InitMTR | Initiate Meter Read by Meter Number Outbound Message |
| D3-RTSender | Real Time Sender |
| D3-RTSnd | Real-time Sender (Landis+Gyr) |
| D3_CIMODR | L+G CIM On Demand Read |
| D3_CIMPING | L+G CIM Ping |
| D3_SCHDEMRES | L+G Schedule Demand Reset |

**Note**: The following apply to all of the above message senders:


Copyright © 2026, Oracle and/or its affiliates.

**Main Tab**:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

**Context Tab**:

- **HTTP Header**: SOAPAction: http://xmlns.oracle.com/ouaf/multispeak_3.0/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/D3/**<SERVICE>**/**<SERVICE>**

  where:

  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
  - **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

## Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

| Outbound Message Type | Description |
|---|---|
| D3-ADDMTRINV | Add Meter to Inventory |
| D3-COMMS | Commission Device |
| D3-CONNECT | Connect Device |
| D3-DECOMMS | Decommission |
| D3-DEMRESET | Demand Reset |
| D3-DERDEV | Deregister Device |
| D3-DISCONNEC | Disconnect Device |
| D3-DVCSTSCHK | Device Status Check |
| D3-INITMRN | Initiate Meter Read by Meter Number |
| D3-INITMTR | Initiate Meter Read by Meter Number |
| D3-MTRADDNOT | Meter Add Notification Outbound Message Type |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Type | Description |
|---|---|
| D3-MTREX | Meter Exchange Notification OB MSG |
| D3-MTRRMV | Meter Remove Notification |

**Note**: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

## External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the L+G Command Center must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

**External System - Landis+Gyr**:

- **External System**: LG
- **Description**: Landis+Gyr
- **Outbound Message Types**:

| Outbound Message Type | Description | Message Sender |
|---|---|---|
| D3-ADDMTRINV | Add Meter to Inventory | Message sender associated with the Add Meter to Inventory Outbound Message Type |
| D3-COMMS | Commission Device | Message sender associated with the Commission Device Outbound Message Type |
| D3-CONNECT | Connect Device | Message sender associated with the Connect Device Outbound Message Type |
| D3-DECOMMS | Decommission | Message sender associated with the Decommission Device Outbound Message |
| D3-DEMRESET | Demand Reset | Message sender associated with the Demand Reset Outbound Message Type |
| D3-DERDEV | Deregister Device | Message sender associated with the Deregister Device Outbound Message Type |
| D3-DISCONNEC | Disconnect Device | Message sender associated with the Disconnect Device Outbound Message Type |
| D3-DVCSTSCHK | Device Status Check | Message sender associated with the Device Status Check Outbound Message Type |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Type | Description | Message Sender |
|---|---|---|
| D3-INITMRN | Initiate Meter Read by Meter Number | Message sender associated with the Initiate Meter Read By Meter Number Outbound Message Type |
| D3-INITMTR | Initiate Meter Read by Meter Number | Message sender associated with the Initiate Meter Read By Meter Number Outbound Message Type |
| D3-MTRADDNOT | Meter Add Notification Outbound Message Type | Message sender associated with the Meter Add Notification Outbound Message Type |
| D3-MTREX | Meter Exchange Notification OB MSG | Message sender associated with the Meter Exchange Notification Outbound Message Type |
| D3-MTRRMV | Meter Remove Notification | Message sender associated with the Meter Remove Notification Outbound Message Type |

Note: The following apply to the above outbound message types*:

- **Processing Method**: Real-time
- **Message XSL**: D3-Request xsl (*use D3-DRRequest.xsl for the Demand Reset command)
- **Response XSL**: D3-Response.xsl (*use D3-DRResponse.xsl for the Demand Reset command)

## Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the L+G Command Center must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

**Service Provider - Landis+Gyr**:

- **Service Provider**: LG
- **Description**: Landis+Gyr
- **External Reference ID**: L+G
- **External System**: Landis+Gyr
- **Out Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID


Copyright © 2026, Oracle and/or its affiliates.

## Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the L+G service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

### Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

### Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

### UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

### Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the L+G service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

| Command | Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|---|
| Device Commission | Device Registration | D3-AddMeterToInventoryMultiSp | Add Meter to Inventory |
| Device Commission | Device Installation | D3-MeterExNotificationMultiSp | Meter Exchange Notification |
| Device Decommission | Device Removal | D3-MtrRmvNotifMultiSpeak | Decommission<br>Meter Remove Notification |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|---|
| Device Deregistration | Device Deregistration | D3-MeterRetireNotification | Deregister Device |
| On-Demand Read (Scalar), Multispeak and CIM | On-Demand Read (Scalar) | MS: D3-InitiateMRByMtrNbr<br>CIM: D3-CIMMeterOnDemandRead | MS: Initiate Meter Read by Meter Number<br>CIM: CIM On Demand Read |
| On-Demand Read (Interval) | On-Demand Read (Interval) | D3-CIMGetLPData | CIM On Demand Read |
| Demand Reset | Demand Reset | D3-ScheduleDemandReset | Demand Reset |
| Device Status Check | Device Status Check | D3-CIMPing | Device Status Check |
| Remote Connect | Remote Connect | D3-InitiateConnectDisconnect | Connect Device |
| Remote Disconnect | Remote Disconnect | D3-InitiateConnectDisconnect | Disconnect Device |

## Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Landis+Gyr endpoint override DVM (D3–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Landis+Gyr adapter and the command used with each:

| DVM Key | Commands |
|---|---|
| CIMService | On-Demand Read (Scalar) - CIM<br>On-Demand Read (Interval) - CIM<br>Device Status Check |
| CD_CB | Remote Connect<br>Remote Disconnect |
| Metering | Device Commissioning (Registration) |


Copyright © 2026, Oracle and/or its affiliates.

| DVM Key | Commands |
|---|---|
| MR_CB | Device Commissioning<br>Device Decommissioning<br>Device Commissioning (Installation)<br>Device Deregistration<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| LGProcessCallout | User Exit Functions |

To define an override Endpoint URI for the Landis+Gyr adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D3–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list). The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.

4. Click the **Add Domain Values** icon (“+”) to add a new key.

   The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.

6. Click **OK**.

   The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.

8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

   Click **Discard** to discard your changes.

   Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

## Configuring Landis+Gyr Extendable Lookups


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Landis+Gyr.

This section outlines some of the extendable lookups that must be configured for use with the Landis+Gyr adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

#### CIM Response Status Extendable Lookup

The CIM Response Status extendable lookup is used to map descriptions to response status codes received from the L+G Command Center. Each value defined for the CIM Response Status extendable lookup should include the following:

- **Response Status**: The CIM status code for the response status
- **Description**: A description of the response status
- **Status**: The status of the lookup value (can be Active or Inactive)

#### CIM Data Source Extendable Lookup

The CIM Data Source extendable lookup is used to map descriptions to data sources defined in the L+G Command Center. Each value defined for the CIM Data Source extendable lookup should include the following:

- **Data Source**: The CIM code for the data source
- **Description**: A description of the data source
- **Status** : The status of the lookup value (can be Active or Inactive)

#### L+G CIM 2.0 Header Verb Lookup

The L+G CIM 2.0 Header Verb Lookup (D3-CIM2HeaderVerbLookup) extendable lookup defines values for message type verbs used in CIM 2.0 messages. Refer to the **International Standard document** for detailed information about verbs. This extendable lookup is pre-populated with values from the International Standard document.

#### L+G CIM 2.0 - Measuring Period Lookup

The L+G CIM 2.0 - Measuring Period Lookup (D3-CIMMeasuringPeriodLookup) extendable lookup defines mappings between measuringPeriod codes and Interval Size values. The values are defined in the **International Standard document**.

#### L+G CIM 2.0 Reply Code Lookup

The L+G CIM 2.0 Reply Code Lookup (D3-CIMReplyCodeLookup) extendable lookup contains the enumeration of the values in the Reply / Error / Code element in replies from the head end system. The values are defined in the **International Standard document** . Values comprise two numbers separated by a dot. Example: `ReplyCode = <ReplyCode Category> “.” <ReplyCode Index>`

This extendable lookup is pre-populated from the International Standard document.

#### L+G CIM 2.0 Reply Result Lookup

The L+G CIM 2.0 Reply Result Lookup (D3-CIMResponseReplyRsltLookup) extendable lookup contains the enumeration of the values for the `Reply.Result` value returned from the head end


Copyright © 2026, Oracle and/or its affiliates.

system. This extendable lookup is pre-populated with values from the **International Standard document**.

#### Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the "Standard Event Name" extendable lookup.

#### UOM Code to Standard UOM Mapping

Usage received from a utility's head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.

#### Interval Status Code to Condition Mapping

Interval usage received from the Landis+Gyr Command Center can include interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Landis+Gyr Interval Status Code to Condition Mapping extendable lookup is used for this purpose. Each value defined for the Landis+Gyr Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The interval status code
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.
- **Description**: A description of the interval status code.
- **Daylight Savings Time Effect for Interval**: Flag that indicates (yes or no) if the status code indicates data during Daylight Saving Time (DST). To ensure that all DST scenarios are handled appropriately, set this parameter to "Yes" for all status conditions that indicate the interval was during DST. This should include mapping an Interval Status of "10" to the "Regular" condition.

## Using the Landis+Gyr Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr (LG) includes a test harness that can be configured to simulate the Landis+Gyr Gridstream Command Center head-end system for testing the two-way commands. The test harness is Multispeak 3.0 standard compliant and includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Middleware Test Harnesses for more information.


Copyright © 2026, Oracle and/or its affiliates.

## Landis+Gyr Interval Data Mapping

This section describes how data in Landis+Gyr import files is mapped to the XML document format created by OSB and sent to Oracle Utilities Smart Grid Gateway.

When Landis+Gyr data is processed, it is initially received in a tilda-separated file format, which is converted into a "plan" XML format before being converted into the "result" XML format which is sent to the IMD Seeder and/or Device Event Seeder inbound services.


- Non-Interval Usage with Additional Fields
- XML 'Plain' XML Format
- Non-Interval Usage to 'Plain' XML Mapping
- 'Plain' XML to IMD Mapping
- Non-Interval 'Plain' XML to IMD Mapping
- Mapping Additional Fields


### Non-Interval Usage with Additional Fields

The following is a sample file of non-interval usage that contains additional fields.

ID~PremiseID~ESIID~Provisioned~Meter~kWh~DateTime~Peak~PeakDateTime~Dmd~TouA~TouB~TouC~TouD~TouE~Volts~PF~PrevKWh~KWhRecv

EMED01~SL002~~~96968280~34315.000~08042010120000AM~2.66~03122009063000AM~~34315.000~0.000~0.000~0.000~0.000~~~34285.000~30

EMED01~SL001~~~96968285~33693.000~08042010120000AM~2.62~03122009061500AM~~33693.000~0.000~0.000~0.000~0.000~~~33653.000~40

This data is mapped to the "plain" XML format.

### XML 'Plain' XML Format

The "Plain" XML contain elements to hold the extra fields (highlighted in bold).

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
           targetNamespace=" http://xmlns.oracle.com/LandisGyrUsage"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MeterReads">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MeterRead">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Origin"/>
              <xs:element name="ServProvExtRefId"/>
              <xs:element name="RecordType">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
```



Copyright © 2026, Oracle and/or its affiliates.

```
                        <xs:enumeration value="MEPMD01" />
                        <xs:enumeration value="EMED01" />
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                <xs:element name="RecordVersion" minOccurs="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="20080519" />
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                <xs:element name="TimeStamp" />
                <xs:element name="Premise" minOccurs="0" />
                <xs:element name="ESIID" minOccurs="0" />
                <xs:element name="Provisioned" minOccurs="0" />
                <xs:element name="MeterID" />
                <xs:element name="Purpose" minOccurs="0" />
                <xs:element name="Comodity" minOccurs="0" />
                <xs:element name="Units" minOccurs="0" />
                <xs:element name="CalcConst" minOccurs="0"/>
                <xs:element name="Interval" minOccurs="0"/>
                <xs:element name="Count" minOccurs="0"/>
                <xs:element name="FirstIntervalDateTime" />
                <xs:element name="Data">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Row" maxOccurs="unbounded"
minOccurs="0">
                        <xs:complexType>
                          <xs:attribute name="v" />
                          <xs:attribute name="s" />
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="kWh" minOccurs="0" />
                <xs:element name="Peak" minOccurs="0" />
                <xs:element name="PeakDateTime" minOccurs="0" />
                <xs:element name="Dmd" minOccurs="0" />
                <xs:element name="TouA" minOccurs="0" />
                <xs:element name="TouB" minOccurs="0" />
                <xs:element name="TouD" minOccurs="0" />
                <xs:element name="TouC" minOccurs="0" />
                <xs:element name="TouE" minOccurs="0" />
                <xs:element name="Volts" minOccurs="0" />
                <xs:element name="PF" minOccurs="0" />
                <xs:element name="ExtraFields" minOccurs="0">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="ExtraField" maxOccurs="255"
minOccurs="0">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="FieldName" minOccurs="0"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                        <xs:element name="FieldValue" minOccurs="0"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="RawData"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

## Non-Interval Usage to 'Plain' XML Mapping

The following table shows the mapping between fields in incoming non-interval data and child elements of MeterReads/MeterRead element in the "Plain" XML format:

| L+G Interval Structure Field | "Plain" XML element |
|---|---|
| ID | RecordType |
| Premise ID | Premise |
| ESIID | ESIID |
| Provisioned | Provisioned |
| Meter | MeterID |
| kWh | kWh |
| Date/Time of initial Read | FirstIntervalDateTime |
| Peak | Peak |
| Peak Date/Time | PeakDateTime |
| Dmd | Dmd |
| TouA | TouA |
| TouB | TouB |
| TouC | TouC |
| TouD | TouD |
| TouE | TouE |
| Volts | Volts |
| PF | PF |
| `<extraFieldName1>` | ExtraFields/ExtraField/FieldName with value of `<extraFieldName1>` |
| | ExtraFields/ExtraField/FieldValue with value in `<extraFieldName1>` |
| `<extraFieldName2>` | ExtraFields/ExtraField/FieldName with value of `<extraFieldName2>` |
| | ExtraFields/ExtraField/FieldValue with value in `<extraFieldName2>` |


Copyright © 2026, Oracle and/or its affiliates.

| L+G Interval Structure Field | "Plain" XML element |
|---|---|
| <extraFieldName3> | ExtraFields/ExtraField/FieldName with value of <extraFieldName3> |
| | ExtraFields/ExtraField/FieldValue with value in <extraFieldName3> |
| <extraFieldNameN> | ExtraFields/ExtraField/FieldName with value of <extraFieldNameN> |
| | ExtraFields/ExtraField/FieldValue with value in <extraFieldNameN> |
| RawData | A record content from incoming file. |

## 'Plain' XML to IMD Mapping

The following table outlines how data from the "plain" XML format is mapped to the InitialLoadIMD format when the Landis+Gyr record type is set to "MEPMD01" (interval usage).

| "Plain" XML element | InitialLoadIMD element | Note |
|---|---|---|
| Record Type | N/A | |
| RecordVersion | N/A | |
| Premise ID | N/A | |
| ESIID | N/A | |
| Provisioned | N/A | |
| Meter ID | dvcIdN | as is |
| Purpose | N/A | |
| Commodity | N/A | |
| Units | externalUOM | as is |
| CalcConst | mcm | as is |
| Interval | spi | Transform from DDHHMM to SPI |
| Count | N/A | |
| FirstIntervalDateTime | stDt | Convert to OUAF date/time format |
| Data | msrs | Value -> msrs/mL/q |
| Status -> msrs/mL/sts/stsL/st | | |
| N/A | imdType | 'D1IL |
| Origin | externalId | the origin attribute in "Plain" XML (incoming file name) |
| N/A | serviceProviderExternalId | L+G |
| RawData | rawData | Vendor-specific "raw" data |

## Non-Interval 'Plain' XML to IMD Mapping


Copyright © 2026, Oracle and/or its affiliates.

The following table outlines how non-interval data in the "plain" XML format is mapped to the InitialLoadIMD format when the Landis+Gyr record type is set to "EMED01" (non-interval usage).

| "Plain" XML Element | InitialLoadIMD element | Notes |
|---|---|---|
| RecordType | N/A | |
| Premise | N/A | |
| ESIID | N/A | |
| Provisioned | N/A | |
| MeterID | dvcIdN | as is |
| kWh | enQty | as is |
| | uom | KWH |
| TimeStamp | enDt | Convert to OUAF date/time format<br>This should be mapped to enDt. Relevant only to non-interval data. |
| Peak | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KW |
| PeakDateTime | enDt | Convert to OUAF date/time format |
| Dmd | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KW and 'Dmd' in "<mcIdN> |
| TouA | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate A' in "<mcIdN> |
| TouB | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH' and 'kWh Rate B' in "<mcIdN> |
| TouC | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate C' in "<mcIdN> |
| TouD | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate D' in "<mcIdN> |
| TouE | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with KWH and 'kWh Rate E' in "<mcIdN> |
| Volts | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with Volts |


Copyright © 2026, Oracle and/or its affiliates.

| "Plain" XML Element | InitialLoadIMD element | Notes |
|---|---|---|
| PF | enQty | as is<br>A separate IMD Seeder will be created with <uom> in IMD Seeder populated with PF |
| N/A | imdType | D1IL |
| Origin | externalId | the origin element in "Plain" XML (incoming file name) |
| N/A | serviceProviderExternalId | L+G<br>This is a constant. There is no such field in incoming structure. |
| RawData | rawData | Vendor-specific "raw" data<br>When the "populateRawIMD" parameters in the EnvironmentSettings.xq file is set to true. |

## Mapping Additional Fields

Measurement and device event files received from the Landis+Gyr head-end system can include additional fields containing data to be imported into Oracle Utilities Smart Grid Gateway. These additional fields must be mapped to elements within the XML document processed by OSB and sent to Smart Grid Gateway.

This mapping can be performed through use of a custom XQuery document, specified in the EnvironmentSettings.xq file via the "modifyResultXMLInput" parameter.

The following sample XQuery documents illustrate how additional fields can be mapped into the XML format sent to Smart Grid Gateway.

### Sample XQuery — Initial Measurements

The following XQuery is an example that shows a transformation that passes in a root element with 3 children (the “result” XML, the “plain” XML, the environment settings) that returns a modified “result” XML For testing purposes, it changes the original value in the <enQty> “result” element and replaces it with a value from the “plain” XML depending on an environment setting variable. The <serviceProviderExternalId> value was also replaced by a hard-coded value.

```
declare namespace lan = "http://xmlns.oracle.com/LandisGyrUsage";
declare namespace xf = "http://tempuri.org/D3/lgimd";
declare namespace soap = "http://schemas.xmlsoap.org/soap/envelope/";
declare function xf:modifyResultXML($modifyResultXMLInput as element(*)) as
element(*){
<InitialLoadIMDList>
{
  for $InitLoadIMD in $ modifyResultXMLInput/InitialLoadIMDList/InitialLoadIMD
  return
    <InitialLoadIMD>
      <preVEE>
        <dvcIdN>{ data($InitLoadIMD/preVEE/dvcIdN) }</dvcIdN>
        <externalId>{ data($InitLoadIMD/preVEE/externalId)}</externalId>
        <uom>{ data($InitLoadIMD/preVEE/uom)}</uom>
        <mcIdN>{ data($InitLoadIMD/preVEE/mcIdN)}</mcIdN>
        <enDt>{ data($InitLoadIMD/preVEE/enDt)}</enDt>
        {
```


Copyright © 2026, Oracle and/or its affiliates.

```
        if ($modifyResultXMLInput/EnvironmentSettings/test1="true")
        then <enQty>{ data($modifyResultXMLInput/lan:MeterReads/lan:MeterRead/
lan:ExtraFields/lan:ExtraField[lan:FieldName
= 'EF4']/lan:FieldValue) }</enQty>
         else <enQty>{ data($modifyResultXMLInput/lan:MeterReads/
lan:MeterRead/lan:ExtraFields/lan:ExtraField[lan:FieldName= 'EF2']/
lan:FieldValue) }</enQty>
         }
       <imdType>{ data($InitLoadIMD/preVEE/imdType) }</imdType>
      </preVEE>
      <serviceProviderExternalId>NewSPId</serviceProviderExternalId>
    </InitialLoadIMD>}</InitialLoadIMDList>
};
declare variable $modifyResultXMLInput as element(*)external;
xf:modifyResultXML($modifyResultXMLInput)
```

### Sample XQuery — Device Events

The following XQuery is an example that shows a transformation that passes in a root element with 3 children (the "result" XML, the "plain" XML, and the environment settings) and returns a modified "result" XML For testing purposes, it changes the original value in the `<externalCommunicationModuleIdentifier>` "result" element and replaces it with a value from the "plain" XML depending on an environment setting variable. The `<externalServiceLocationId>` value is also replaced by a hard-coded value.

```
declare namespace lan = "http://xmlns.oracle.com/LandisGyrEvent";
declare namespace xf = "http://tempuri.org/D3/event";
declare namespace soap = "http://schemas.xmlsoap.org/soap/envelope/";
declare function xf:modifyResultXML($modifyResultXMLInput as element(*)) as
element(*){
  <DeviceEventSeeder>
    <externalSenderId>{data($modifyResultXMLInput/DeviceEventSeeder/
externalSenderId) }</externalSenderId>
    <deviceIdentifierNumber>{ data($modifyResultXMLInput/DeviceEventSeeder/
deviceIdentifierNumber) }</deviceIdentifierNumber>
    <externalEventName>{ data($modifyResultXMLInput/DeviceEventSeeder/
externalEventName) }</externalEventName>
    <eventDateTime>{ data($modifyResultXMLInput/DeviceEventSeeder/
eventDateTime) }</eventDateTime>
    <externalSourceIdentifier>{ data($modifyResultXMLInput/DeviceEventSeeder/
externalSourceIdentifier)
}</externalSourceIdentifier>
    <eventInformation>
      <externalEventCategory>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalEventCategory)
}</externalEventCategory>
      <externalEventSeverity>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalEventSeverity)
}</externalEventSeverity>
      <externalDeviceType>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalDeviceType)
}</externalDeviceType>
      <externalServiceLocationId>{1234 }</externalServiceLocationId>
      {
      if ($modifyResultXMLInput/EnvironmentSettings/testA="true")
```


Copyright © 2026, Oracle and/or its affiliates.

```
      then
<externalCommunicationModuleIdentifier>{ data($modifyResultXMLInput/
lan:DeviceEvents/lan:DeviceEvent/lan:DeviceType)}</
externalCommunicationModuleIdentifier>
      else <externalCommunicationModuleIdentifier>{data($modifyResultXMLInput/
lan:DeviceEvents/lan:DeviceEvent/lan:CategoryId)}</
externalCommunicationModuleIdentifier>
      }
     <externalStatusValue>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalStatusValue)
}</externalStatusValue>
      <externalStatusDateTime>{data($modifyResultXMLInput/DeviceEventSeeder/
eventInformation/externalStatusDateTime)}</externalStatusDateTime>
    </eventInformation>
  </DeviceEventSeeder>
};
declare variable $modifyResultXMLInput as element(*) external;
xf:modifyResultXML($modifyResultXMLInput)
```

## MV90 for Itron

The Oracle Utilities Smart Grid Gateway MV-90 Adapter for Itron uses Oracle Service Bus (OSB) to facilitate communication between Oracle Utilities Smart Grid Gateway and MV-90. The following functionality is included:

**Measurement Data Loading** - Data parsing and transformation from MV-90 binary format into the Oracle Utilities Service and Measurement Data Foundation unified format.

**Measurement Data Processing** - Configurable mapping for MV-90 status codes to Oracle Utilities Service and Measurement Data Foundation standard values, along with configurable device event creation based on MV-90 status codes.

The following table describes the attributes of the adapter:

| Attribute | Details |
|---|---|
| Currently Supported Version | SGG supports the .mv9 binary, mainframe data format. |
| Protocol | Binary file format. |
| Market(s) | Worldwide, but largely USA. |
| Architecture | AMR |

- MV90 Adapter Processing
- Configuring an MV90 Head-End System
- Configuring MV90 Extendable Lookups

### MV90 Adapter Processing

This section provides details concerning the OSB processing and OUAF objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data Loading


Copyright © 2026, Oracle and/or its affiliates.

## Initial Measurement Data Loading

The initial measurement data load and subsequent device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway service format. Payloads contain measurements in a head-end specific format OSB then places each service call into a JMS queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel then a service creates initial measurements with data in a common format with head-end-specific processing as needed.

- Initial Measurements
- Base Package Business Objects

### Initial Measurements

The usage data exported from the AMI head-end system as a file in MV-90 format is loaded into Oracle Utilities as Initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D5-USAGE-BASE** contains components responsible for the actual processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to the SGG-D5-USAGE-CM project.
2. **SGG-D5-USAGE-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D5-USAGE-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRawIMD | Determines if the initial measurement data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElementInterval | Holds the name of inbound web service for the interval IMD seeder. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterUsage | Determines if usage should be filtered. | true<br>false |
| usePhysicalChannel | Optional value. Determines whether the physical channel ID is passed to the IMD seeder to create the measuring component identifier number. If this is set to false, the MV90 LOGCHAN field is used. | true<br>false (default) |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| fieldForDvcIdN | Optional value. Specifies which field is used as the value for the device ID. | DC_RECID<br>DC_CUSTID<br>DC_METERID (default) |
| MV90ScalarChannelSuffix | Optional value. Holds a suffix value to be added to the measuring component identifier number when a scalar IMD is created for register reads. Default value is “_S” | |
| processMV90ScalarData | Optional value. Determines if register reads are processed. | true<br>false (default) |
| usePulseMultiplierAndOffset | Optional value. If set to "true", specifies the pulse values are multiplied by the pulse mutlipler. | true<br>false (default) |

> **Note**
>
> This setting only applies to MV90 files that have a Pulse datatype.

### Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D5-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

### Configuring Initial Measurement Publishing Output


Copyright © 2026, Oracle and/or its affiliates.

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |


Copyright © 2026, Oracle and/or its affiliates.

### Filtering Usage

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **MV90 UOM Code to Standard UOM Mapping** extendable lookup (D5-UOMtoStdUOMCodeMapLookup) are passed into the system for processing.

### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

### Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Initial Measurement Data Load with Register Reads

In the MV90 format, measurement data can include interval data as well as register reads that can be used to create scalar measurements. For information on configuring the adapter for loading scalar data based on register reads, see Override Processing Methods for Scalar Initial Measurement Data.

Base Package Business Objects

The MV-90 Adapter for Itron base package includes the following initial measurement business objects:

| Business Object Name | Description |
|---|---|
| D5-InitialLoadIMDInterval | MV-90 Initial Load IMD |
| D5-InitialLoadIMDScalar | MV90 - Initial Load IMD - Scalar |

## Configuring an MV90 Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway MV-90 Adapter for Itron to receive usage from an MV-90 system.

- Inbound Web Services
- External System
- Service Provider


Copyright © 2026, Oracle and/or its affiliates.

- Processing Methods

## Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including incoming usage.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

| Inbound Web Service Name | Description |
|---|---|
| D1-InitialLoadIMD | IMD Seeder |
| D1-PayloadErrorNotif | Payload Error Notification |
| D1-PayloadStatistics | Payload Statistics |
| D1-PayloadSummary | Payload Summary |

**Note**: The following apply to all of the above inbound web services:

**Message Options**

- **Trace**: No
- **Debug**: No
- **Active**: Yes

**Operations**

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

## External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the MV-90 system must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

**External System - MV-90**:

**External System**: MV90

**Description**: MV90 - Mainframe File Format

## Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that


Copyright © 2026, Oracle and/or its affiliates.

receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the MV-90 system must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about creating service providers.

**Service Provider - MV-90**:

- **Service Provider**: MV90
- **Description**: Head-End System for MV90
- **External Reference ID**: MV90
- **External System**: MV90 - Mainframe File Format
- **Out Name/ID in Their System**: MV90
- **AMI Device ID Type**: Serial Number
- **AMI Measuring Component ID Type**:

## Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the MV90 service provider. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about configuring processing methods.

### Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

### Override Processing Methods for Scalar Initial Measurement Data

In the MV-90 format, meter read start and stop values can represent either a calculated read based on the interval data that is included in the MV90 data, or a register read that has been logged using a separate channel. When the values represent a register read the application can create a second file for the scalar initial measurements. The creation of a scalar initial measurement file is triggered by the Register Type flag, DC_REGTYPE, in the MV-90 data and the processMV90ScalarData variable in the EnvironmentSettings.xq file. The following register read values are valid for the DC_REGTYPE field:

| Value | Description |
|---|---|
| V | Visual energy readings |
| D | Visual demand and energy readings |
| E | Encoder register readings (energy only). |
| B | Both encoder and visual energy readings. |


Copyright © 2026, Oracle and/or its affiliates.

| Value | Description |
|---|---|
| C | Calculate stop meter readings from the encoder base reading |

In order to configure your adapter to create scalar initial measurements based on register reads, first define new measuring component types and register measuring components. Then use the new measuring component types to specify override processing methods on the Initial Measurement Creation processing method for your service provider.
**Measuring Component Types for Scalar Reads**: To create initial measurement data for register reads, first create new measuring component types to represent the registers. These will be used to create measuring components for scalar reads.

**Example Measuring Component Type**

- **Measuring Component Type:** MV90_SCALAR_REG
- **Description:** MV90 Scalar Register
- **Measuring Component Business Object:** Register
- **Measurement Business Object:** Measurement
- **Service Type:** Electric Service
- **Allow Negative Consumption:** Allowed
- **Consumptive/Subtractive:** Subtractive
- **Read Method:** Automatic Read

Other attributes such as Value Identifiers, VEE Groups, and so on, should be defined based on requirements

**Register Measuring Components for Scalar Reads**: Use the new measuring component types to create measuring components for the scalar measurements created by the register read process. You must create a corresponding register measuring component for each interval channel for which you will receive scalar reads. The channel number for the registers must be based on the channel numbers of the corresponding interval channels, plus the suffix defined in the “MV90ScalarChannelSuffix” environment setting. The default value for this setting is “_S”. These registers must also be added to the device configuration used by the interval channels (in other words, both measuring components - interval and scalar - must be on the same device). The following table shows examples of these values for the measuring components:

| Device Type/Serial Number | MV-90 Interval Channel Number | MV-90 Scalar Register Channel Number |
|---|---|---|
| MV90 Electric/00001 | 1 | 1_S |
| MV90 Electric/00002 | 1 | 1_S |
| MV90 Electric/00003 | 1 | 1_S |

**Override Processing Methods for MV-90 Service Provider**: Add an override processing method to the Initial Measurement Creation processing method for the MV90 service provider for each register measuring component type. The business object for each override processing method should be “D5-InitialLoadIMDScalar”. The following tables shows examples of the override processing methods:


Copyright © 2026, Oracle and/or its affiliates.

| Measuring Component Type | Business Object |
|---|---|
| MV90 Scalar Register | D5-InitialLoadIMDScalar |
| Measuring Component Type 2 | D5-InitialLoadIMDScalar |
| Measuring Component Type n | D5-InitialLoadIMDScalar |

#### Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

#### UOM Translation

UOM translation processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map MV90 UOM codes to standard UOM codes when receiving usage from the MV-90 system.

## Configuring MV90 Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway MV90 Adapter for Itron.

This section outlines some of the extendable lookups that must be configured for use with the MV-90 Adapter for Itron. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

#### Interval Status Code to Event Mapping

The MV-90 Adapter for Itron does not accept device events from an MV-90 system, but can create device events based on specific interval status codes in an incoming usage reading. The MV-90 Interval Status Code to Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when creating device events based on interval status codes received with usage from the MV-90 system.

Each value defined for the Interval Status Code to Event Mapping extendable lookup should include the following:

- **MV90 Interval Status**: The interval status code used by the MV-90 system
- **Description**: A description of the MV-90 interval status code
- **Event Duration Mode**: The duration type for the event. Can be either “Individual” or “Continuous” (used for events with a duration, such as an outage).
- **Device Event Type**: The Device Event Type for the device event created for this interval status code. For status codes with an Event Duration Mode of “Continuous” this is the start event, or the first of the paired events created for this status code.


Copyright © 2026, Oracle and/or its affiliates.

- **End Event Type**: The Device Event Type for the “end” device event created for this interval status code. For status codes with an Event Duration Mode of “Continuous” this is the end event, or the last of the paired events created for this status code.

**Example**: The MV-90 “Low Voltage” status code could be configured to create a “Low Voltage” device event as follows:

- **MV90 Interval Status**: <MV-90 interval status code for low voltage>
- **Description**: Low Voltage detected for meter
- **Event Duration Mode**: Individual
- **Device Event Type**: Low Voltage
- **End Event Type**: N/A

**Example**: The MV-90 “Power Outage” status code could be configured to create a “Last Gasp” device event, and a “Power Restoration” event as follows:

- **MV90 Interval Status**: <MV-90 interval status code for outage>
- **Description**: Outage
- **Event Duration Mode**: Continuous
- **Device Event Type**: Last Gasp
- **End Event Type**: Power Restoration

#### UOM Code to Standard UOM Code Mapping

Usage received from a utility’s head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.

#### Other Extendable Lookups

| Business Object Name | Description |
|---|---|
| D5 - ChanelStatusCodeLookup | MV-90 Channel Status Code Lookup |
| D5-IntStsCdToCndMapPrLookup | Condition Mapping with Priority |

## Networked Energy Services

The Oracle Utilities Smart Grid Gateway Adapter for Networked Energy Services (NES) supports two-way communication with the NES server. Communications include measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, and on-demand read. The adapter uses Oracle Service Bus (OSB) and Oracle Business Process Execution Language Process Manager (BPEL PM) to facilitate communication between Oracle Utilities Smart Grid Gateway and NES. The following table describes the attributes of the adapter:


Copyright © 2026, Oracle and/or its affiliates.

| Attribute | Details |
|---|---|
| Currently Supported Version | Networked Energy Services 5.3 |
| Protocol | Proprietary (NES specific) |
| Market(s) | Worldwide |
| Architecture | ANSI and IEC, PLC |

- Networked Energy Services Adapter Processing
- Configuring a Networked Energy Services Head-End System
- Configuring Networked Energy Services Extendable Lookups
- Configuring NES Usage and Event Extract Processing

## Networked Energy Services Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

### Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements
- Device Events
- Base Package Business Objects

#### Initial Measurements

Usage data is exported from the head-end system as a file in NES format and is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D4-USAGE-BASE** contains components responsible for the actual processing of incoming data. This should not be modified during configuration.
2. **SGG-D4-USAGE-CM** allows you to segment your customizations so that future upgrades of base functionality implemented in SGG-D4-USAGE-BASE do not affecting the customization and environment settings.

The runtime configuration settings for the SGG-D4-USAGE-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data


Copyright © 2026, Oracle and/or its affiliates.

processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the initial measurement data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElementInterval | Holds the name of inbound web service for the interval IMD seeder. | |
| destinationRootElementScalar | Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterUsage | Determines if usage should be filtered. | true<br>false |

### Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

#### Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D4-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

#### Configuring Initial Measurement Publishing Output


Copyright © 2026, Oracle and/or its affiliates.

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |


Copyright © 2026, Oracle and/or its affiliates.

### Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **NES UOMCode Mapping Extendable Lookup** extendable lookup (D4-HeadendUOMLookup) are passed into the system for processing.

### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

### Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

## Device Events

The device event data exported from the head-end system as a file in NES format is loaded into Oracle Utilities as a Device Event. One of your configuration tasks is to customize the device events processing.

The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D4-EVENT-BAS**E contains components responsible for the actual processing of incoming data. This should not be modified during configuration.
2. **SGG-D4-EVENT-CM** allows you to segment your customizations so that future upgrades of base functionality implemented in SGG-D4-EVENT-BASE do not affecting the customization and environment settings.

The runtime configuration settings for the SGG-D4-EVENT-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the device event data is populated as raw data. | true<br>false |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElement | Holds the name of inbound web service for the device event seeder. | |
| serviceProviderExternalId | The External ID of the NES service provider. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |

### Publishing Events

The NES adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D4-EVENT-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

### Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.


Copyright © 2026, Oracle and/or its affiliates.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |

### Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **NES Device Event Mapping** extendable lookup (D4-DeviceEventMappingLookup) are passed into the system for processing.

### Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device**


Copyright © 2026, Oracle and/or its affiliates.

**Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D4-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

## Base Package Business Objects

The NES adapter base package includes the following device, initial measurement, and device event business objects:

| Business Object Name | Description |
|---|---|
| D4-InitialLoadIMDInterval | NES Initial Load IMD Interval |
| D4-InitialLoadIMDScalar | NES Initial Load MD Scalar |


Copyright © 2026, Oracle and/or its affiliates.

| Business Object Name | Description |
|---|---|
| D4-SmartMeter | Smart Meter - NES |
| D4-EventExtractScheduler | NES Event Extract Scheduler |
| D4-PayloadExtractScheduler | Payload Extract Scheduler |
| D4-UsageExtractScheduler | NES Usage Extract Scheduler |

## Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes and Web Services

### Communication Flows

The table below lists the communications created for each NES command:

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Remote Connect | Retrieve Meter Information<br>Read Load Status<br>Read Billing Data On-Demand<br>Connect Load | Read Load Status Notification<br>Read Billing Data On-Demand Notification<br>Connect Load Notification | Create IMD<br>Connect Device |
| Remote Disconnect | Disconnect Load<br>Read Billing Data On-Demand | Disconnect Load Notification<br>Read Billing Data On-Demand Notification | Create IMD<br>Disconnect Device |
| Device Commissioning | Add Meter Request<br>Set ATM Configuration | Set ATM Configuration Notification | Device Commissioning |
| Device Decommissioning | Decommission Meter Request | Decommission Meter Notification | Device Decommissioning |
| On-Demand Read (Scalar) | Read Billing Data On-Demand | Read Billing Data On-Demand Notification | Create IMD |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| On-Demand Read (Interval) | Read Load Profile On Demand | Read Load Profile On-Demand Notification | Create IMD |

### Commissioning and Decommissioning Communications

Commissioning a previously decommissioned NES device requires a different process than commissioning a new device.

When you commission a device for the first time, a commissioning command is sent to BPEL to create the meter in NES using the DeviceManager.CreateMeter command. If Automatic Topology Management is selected then the DeviceManager.SetATMConfiguration command is also sent.

When you decommission a device, BPEL sends the DeviceManager.Move command to NES for the device. In NES you need to confirm that the device has been moved to a non-DC tree. You then need to manually delete the device from the non-DC tree. To permanently decommission the device you must delete it by using the DeviceManager.Delete command from the NES interface.

To recommission a device that has been previously decommissioned, you need to log into NES and manually move the meter from the non-DC tree to the DC tree that it belonged to previously. This will not be possible if the device has been deleted from within NES.

## Device Communication Base Package Business Objects

The NES base package includes the following communication business objects:

### Meter Commissioning

| Business Object | Description |
|---|---|
| D4-AddMeterRequest | Add Meter Request |
| D4-GenericAMIDeviceIdentifier | Generic AMI Device Identifier |
| D4-RetrieveMeterIdentifier | Retrieve Meter Identifier |
| D4-RetrieveMeterInfo | Retrieve Meter Information |
| D4-SetATMConfigNotification | Set ATM Configuration Notification |
| D4-SetATMConfiguration | Set ATM Configuration |

### Meter Decommissioning

| Business Object | Description |
|---|---|
| D4-DecommMeterNotification | Decommission Meter Notification |
| D4-DecommMeterRequest | Decommission Meter Request |

### On-Demand Read

| Business Object | Description |
|---|---|
| D4-ReadBillingData | Read Billing Data On-Demand |
| D4-ReadBillingDataNtf | Read Billing Data On-Demand Notification |


Copyright © 2026, Oracle and/or its affiliates.

| Business Object | Description |
|---|---|
| D4-ReadLoadProfile | Read Load Profile On Demand |
| D4-ReadLoadProfileNtf | Read Load Profile On-Demand Notification |
| D4-ReadLoadStatus | Read Load Status |
| D4-ReadLoadStatusNtf | Read Load Status Notification |

#### Remote Connect

| Business Object | Description |
|---|---|
| D4-ConnectLoad | Connect Load |
| D4-ConnectLoadNtf | Connect Load Notification |

#### Remove Disconnect

| Business Object | Description |
|---|---|
| D4-DisconnectLoad | Disconnect Load |
| D4-DisconnectLoadNtf | Disconnect Load Notification |

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted.

| Outbound Message Type | Description |
|---|---|
| D4-ADDMETREQ | Add Meter Request |
| D4-CONLOAD | Connect Load |
| D4-DECMETREQ | Decommission Meter Request |
| D4-DISCONLD | Disconnect Load |
| D4-HNLBATEVT | NES - Handle Batch Event |
| D4-HNLBATUSG | NES - Handle Batch Usage |
| D4-RDLOADST | Read Load Status |
| D4-READLP | Read Load Profile |
| D4-READOUT | Read Billing Data |
| D4-RETMETID | Retrieve Meter Identifier |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Type | Description |
|---|---|
| D4-RETRIEVE | Retrieve Meter Information |
| D4-SETATMCFG | Set ATM Configuration |

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway Adapter for NES uses the following inbound web services to import usage and device events:

| Inbound Web Service | Description |
|---|---|
| D1-DeviceEventSeeder | Used for upload of device events.<br>The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event.<br>If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created. |
| D1-InitialLoadIMD | Used for initial measurement upload.<br>The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system. |

The Oracle Utilities Smart Grid Gateway Adapter for NES includes the following inbound web services for receiving communications from the NES:

| Inbound Web Service Name | Description |
|---|---|
| D4-ConnectLoadNotification | Connect Load Notification |
| D4-DecommMeterNotificationXAI | Meter Decommission Notification |
| D4-DisconnectLoadNotification | Disconnect Load Notification |
| D4-ReadBDNotificationXAI | Read Billing Data On-Demand Notification |
| D4-ReadingATMConfigNotification | Get Response Initiate Read By Meter Number |
| D4-ReadLoadStatusNotification | Read Load Status Notification |
| D4-ReadLPNotificationXAI | Read Load Profile On-Demand Notification |


Copyright © 2026, Oracle and/or its affiliates.

### Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway Adapter for NES includes the following message senders:

| Message Sender | Description |
|---|---|
| D4-ADDMETREQ | Add Meter Request |
| D4-CONLOAD | Connect Load |
| D4-DECOMMIS | Device Decommissioning |
| D4-DISCONLD | Disconnect Load |
| D4-HNLBATEVT | NES Handle Batch Event |
| D4-HNLBATUSG | NES Handle Batch Usage |
| D4-RDLOADST | Read Load status |
| D4-READBILLD | NES On-Demand Read (Scalar)(2) |
| D4-READLP | Read Load Profile |
| D4-RETMETID | Retrieve Meter Identifier |
| D4-RETRIEVE | Retrieve Meter Information |
| D4-ReadBill | Read Billing Data |
| D4-SETATMCFG | Set ATM Configuration |

BPEL Processes and Web Services

These processes are responsible for performing the conversion from Oracle Utilities format to NES format, invoking process callouts and invoking the remote endpoint to trigger the device events.

All web services receive XML from an incoming service then transform the call to a format which is recognized by the NES system.

**CreateMeter**: Receives the incoming XML from the message sender service, transforms it to the NES API parameters node, passes it to the DeviceManager.Create NES API web service method then collects XML from CreateMeter to be returned to Oracle Utilities Smart Grid Gateway.

***Messages***

**CreateMeter RequestMessage**: Contains the parameters which are formatted from Oracle Utilities Smart Grid Gateway to match the incoming XML for the NES API DeviceManager.Create method.

**CreateMeter ResponseMessage**: Contains the Device ID that is used when the DeviceManager.Create method validates the security key and the XML parameters successfully.

***Related Processes***

Meter Commissioning

**ConnectLoadRemoteConnect**


Copyright © 2026, Oracle and/or its affiliates.

Receives the incoming XML from the D4-CONLOAD (Connect Load) message sender in Oracle Utilities Smart Grid Gateway and transforms to NES format.

**GrabBatchEvents**
Retrieves the un-processed events from the event repository; transforms each batch to replace GUID identifiers with a readable form, and finally prepares a new file for OSB before deleting the batch of events. The scheduled call to this service can contain the preferred dates to query for or can accept the default which is the previous day usage.

***Messages***

GrabBatchEventsSoapIn

GrabBatchEventsSoapOut

**GrabBatchUsage**
Retrieves the un-processed usage from the usage repository; transforms each batch to replace GUID identifiers with a readable form, and finally prepares a new file for OSB before deleting the batch of usage. The scheduled call to this service can contain the preferred dates to query for or can accept the default which is the previous day usage.

***Messages***

GrabBatchUsageSoapIn

GrabBatchUsageSoapOut

***Related Processes***

Connection

Event Management

Usage Processing

**MeterDecommissioning**
Receives the incoming XML from the message senderOutbound service and transforms to NES API parameters node to be passed to the DeviceManager.Move NES API web service method. The process then collects the return XML to be returned to Oracle Utilities Smart Grid Gateway.

***Messages***

**MeterDecommissioningRequestMessage**: This message is the input for the Move method. The method contains the parameters conveniently formatted from Oracle Utilities Smart Grid Gateway to match the incoming XML for the NES API DeviceManager.Move method.

**MeterDecommissioningResponseMessage**: This message contains status stating that the DeviceManager.Create method validated the security key and the XML parameters are accepted successfully. If the status is SUCCEEDED, it also contains Tracking ID for the call, Device ID and StatusType ID.

***Related Processes***

Meter Decommissioning

**PerformCommand**
This web service facilitates most of the communication between Oracle Utilities Smart Grid Gateway and the NES system. If your implementation requires functionality that was not delivered as part of the base package this web service will play an integral part in managing those customizations.


Copyright © 2026, Oracle and/or its affiliates.

PerformCommand executes a command which generates the events that advance the Oracle Utilities Smart Grid Gateway lifecycle then returns the output string to the caller. The output might be a tracking ID or information about a device.

***Messages***

**PerformCommandSoapIn**: The sXmlParameters input is prepared from within Oracle Utilities Smart Grid Gateway to execute primarily a group of five commands, these are Read Load Status, On-Demand Read, Connect Load, Disconnect Load and Set ATM Configuration.

**PerformCommandSoapOut**:

The return from PerformCommand is the xml from the DeviceManager.PerformCommand for Read Load Status, On-Demand Read, Connect Load, Disconnect Load and Set ATM Configuration.

**ReceivePanoramixEvents**
This web service is actually never called by the implementation, but rather it is called by NES and is needed when configuring the NES server.

ReceivePanoramixEvents obtains the APIKey from the NES authentication service, then, with the event definition ID from the parameter list for this service, switches to the branch for the event definition to begin any specific processing. For example, an on-demand read completion event means that the BPEL PM process needs to retrieve the usage using the RESULTID found in the event xml. All events and usage to be passed on to Oracle Utilities Smart Grid Gateway must first be transformed to replace the GUID with a readable form items. This involves using the BPEL PM DVM lookup functionality.

***Messages***

ReceivePanoramixEventsSoapIn

ReceivePanoramixEventsSoapOut

***Related Processes***

Event Management

**PrePerformCommandCallout / PostPerformCommandCallout**
These web services are hooks designed to allow customization of messages just prior and just after sending them to the NES server. Your implementation may wish to modify the data for special purposes. This is the one of the only places where modification of BPEL PM code is allowed.

**ReceivedPanoramixEventsCallout**
This web service is a hook designed to allow customization of messages just prior and just after sending them to the NES server. Your implementation may wish to modify the data for special purposes. This is one of the only places where modification of BPEL PM code is allowed.

**ReadBillingDataOnDemand**
This web service method, initiate a call to NES system with formatted inbound xml parameter. First it transforms the Parameters received from the calling system in to NES system recognizable format. Then it invokes a call to the NES system's Device Manager Service. When the call is invoked the system receives the return XML from DeviceManager stating the status of the command execution and, if succeeded, the tracking ID of this call.

***Messages***

**ReadBillingDataOnDemandRequestMessage**: Contains the input parameter for DeviceManager Perform Command to read billing data OnDemand. It formats parameters from


Copyright © 2026, Oracle and/or its affiliates.

the calling system to match incoming XML for the NES DeviceManager.PerfomCommand method.

**ReadBillingDataOnDemandResponseMessage**: Transforms the response xml from NES system format to Oracle Utilities Smart Grid Gateway format.

***Related Processes***

On Demand Reads

**ReadLoadProfileOnDemand**
This web service method, initiate a call to NES system's Device Manager Service with formatted inbound xml parameter. First it transforms the Parameters received from the calling system in to NES system recognizable format. Then it invokes a call to the NES system thru Device Manager Service. After invoking the call, the system receives the response message from DeviceManager stating the status of the command execution and, if succeeded, the tracking ID of this call.

***Messages***

**ReadLoadProfileOnDemandRequestMessage**: This message is for interval on demand read request to the NES system. Transforms the incoming XML from Oracle Utilities Smart Grid Gateway to NES recognizable format.

**ReadLoadProfileOnDemandResponseMessage**: Transforms the response xml for ReadLoadProfileOnDemandRequest from NES system format to Oracle Utilities Smart Grid Gateway format.

***Related Processes***

On Demand Reads

**ReadLoadStatusRemoteConnect**
Receives the incoming XML from the D4-RDLOADST ( Read Load Status) message sender in Oracle Utilities Smart Grid Gateway and transforms to NES format.

**RemoteDisconnect**
Receives the incoming XML from the D4-DISCONLD (Disconnect Load) message sender in Oracle Utilities Smart Grid Gateway and transforms to NES format.

**RetrieveMeterInfo**
Initiates a call to the Request Management Service with formatted inbound xml parameters. First it transforms the Parameters received from the calling system in to NES system recognizable format. Then it invokes a call to the NES system thru Request Management Service and retrieves a response with Meter Information.

***Messages***

**RetrieveMeterInformationRequestMessage**: Contains the input parameter for DeviceManager Perform Command to retrieve meter information. It also formats parameters from the calling system to match incoming XML for the NES DeviceManager.PerfomCommand method.

**RetrieveMeterInformationResponseMessage**: Transforms the response xml from NES system format to Oracle Utilities Smart Grid Gateway format.

***Related Processes***

Remote Disconnect

Remote Connect


Copyright © 2026, Oracle and/or its affiliates.

**RetrieveMeterIdentifier**
Receives a call from Oracle Utilities Smart Grid Gateway for RetrieveMeterIdentifier process and invokes the RetrieveByParameter method in the NES system's DeviceManager service.

Converts incoming Oracle Utilities Smart Grid Gateway formatted input to sXmlParameters in NES format then invokes RetrieveMeterIdentifier.

***Messages***

**RetrieveMeterIdentifierRequestMessage**: Transforms the xml received from the calling system to match incoming XML for the NES API DeviceManager.RetrieveByParameter method. This method requires the Type of ID and the associated ID as input.

**RetrieveMeterIdentifierResponseMessage**: When the XML is synchronously returned from the head-end system for the RetrieveByParameter call, the DEVICEID node is only included in the APIPAYLOAD if the status of the call is Succeeded.

***Related Processes***

Meter Commissioning

**SetATMConfiguration**
This web service method is specific to set Automated Topology Management (ATM) configuration command which assigns the Meter to the Data Concentrator.

Converts incoming Oracle Utilities Smart Grid Gateway formatted input to sXmlParameters in NES format then invokes the DeviceManager PerformCommand. A connection should establish to begin the NES command processing.

***Messages***

**SetATMConfigurationRequestMessage**: This message is the input for the SetATMConfiguration method. The method contains the parameters conveniently formatted from Oracle Utilities Smart Grid Gateway to match the incoming XML for the NES API DeviceManager.PerformCommand method where the COMMANID is Constants.DeviceCommands.SET_ATM_CONFIGURATION

**SetATMConfigurationResponseMessage**: when the XML is synchronously returned from the head-end system for the PerformCommand call, the COMMAND node is only included in the APIPAYLOAD if the status of the call is Succeeded.

***Related Processes***

Meter Commissioning

### NES Web Services

The following table describes NES web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

| Smart Grid Gateway Commands | AMI Adapter Business Objects | NES Web Services | NES Operations |
|---|---|---|---|
| Device Commissioning | D4-AddMeterRequest<br>D4-RetrieveMeterIdentifier<br>D4-SetATMConfiguration | DeviceManager | Create<br>RetrieveByParameter<br>PerformCommand:<br>SET_ATM_CONFIGURATION |
| Device Decommissioning | D4-DecommMeterRequest | DeviceManager | PerformCommand:<br>MOVE_DEVICE_ADD |


Copyright © 2026, Oracle and/or its affiliates.

| Smart Grid Gateway Commands | AMI Adapter Business Objects | NES Web Services | NES Operations |
|---|---|---|---|
| Remote Connect | D4-ConnectLoad<br>D4-ReadLoadStatus<br>D4-RetrieveMeterInfo | DeviceManager | Retrieve<br>PerformCommand:<br>CONNECT_LOAD<br>READ_LOAD_STATUS |
| Remote Disconnect | D4-DisconnectLoad | DeviceManager | PerformCommand:<br>DISCONNECT_LOAD |
| On-Demand Read | D4-ReadBillingData | DeviceManager | PerformCommand<br>Command ID:<br>READ_BILLING_DATA_ON_DEMAND (Scalar)<br>READ_LOAD_PROFILE_ON_DEMAND (Interval) |

## Configuring a Networked Energy Services Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway NES Adapter to communicate with the NES software.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the NES in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

| Inbound Web Service Name | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Device Event Seeder |
| D1-InitialLoadIMD | IMD Seeder |
| D1-PayloadErrorNotif | Payload Error Notification |
| D1-PayloadStatistics | Payload Statistics |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service Name | Description |
|---|---|
| D1-PayloadSummary | Payload Summary |
| D4-ConnectLoadNotification | Connect Load Notification |
| D4-DecommMeterNotificationXAI | Meter Decommission Notification |
| D4-DisconnectLoadNotification | Disconnect Load Notification |
| D4-ReadBDNotificationXAI | Read Billing Data On-Demand Notification |
| D4-ReadingATMConfigNotification | Get Response Initiate Meter Read By Meter Number |
| D4-ReadLoadStatusNotification | Read Load Status Notification |
| D4-ReadLPNotificationXAI | Read Load Profile On-Demand Notification |

Note: The following apply to all of the above inbound web services:

**Message Options**

- **Trace**: No
- **Debug**: No
- **Active**: Yes

**Operations**

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

## Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

| Message Sender | Description |
|---|---|
| D4-ADDMETREQ | Add Meter Request |
| D4-CONLOAD | Connect Load |
| D4-DECOMMIS | Device Decommissioning |
| D4-DISCONLD | Disconnect Load |
| D4-HNLBATEVT | NES Handle Batch Event |
| D4-HNLBATUSG | NES Handle Batch Usage |
| D4-RDLOADST | Read Load status |
| D4-READBILLD | NES On-Demand Read (Scalar)(2) |
| D4-READLP | Read Load Profile |
| D4-RETMETID | Retrieve Meter Identifier |
| D4-RETRIEVE | Retrieve Meter Information |


Copyright © 2026, Oracle and/or its affiliates.

| Message Sender | Description |
|---|---|
| D4-ReadBill | Read Billing Data |
| D4-SETATMCFG | Set ATM Configuration |

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction: http://xmlns.oracle.com/ouaf/echelon/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/D4/**<SERVICE>**/**<SERVICE>**

  where:

  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
  - **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

## Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

| Outbound Message Type | Description |
|---|---|
| D4-ADDMETREQ | Add Meter Request |
| D4-CONLOAD | Connect Load |
| D4-DECMETREQ | Decommission Meter Request |
| D4-DISCONLD | Disconnect Load |
| D4-HNLBATEVT | NES - Handle Batch Event |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Type | Description |
|---|---|
| D4-HNLBATUSG | NES - Handle Batch Usage |
| D4-RDLOADST | Read Load Status |
| D4-READLP | Read Load Profile |
| D4-READOUT | Read Billing Data |
| D4-RETMETID | Retrieve Meter Identifier |
| D4-RETRIEVE | Retrieve Meter Information |
| D4-SETATMCFG | Set ATM Configuration |

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

## External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the NES must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

**External System - NES**:

- **External System**: NES
- **Description**: NES
- **Outbound Message Types**::

| Outbound Message Type | Message Sender |
|---|---|
| D4-ADDMETREQ | D4-ADDMETREQ |
| D4-CONLOAD | D4-CONLOAD |
| D4-DECMETREQ | D4-DECOMMIS |
| D4-DISCONLD | D4-DISCONLD |
| D4-HNLBATEVT | D4-HNLBATEVT |
| D4-HNLBATUSG | D4-HNLBATUSG |
| D4-RDLOADST | D4-RDLOADST |
| D4-READLP | D4-READBILLD |
| D4-READOUT | D4-READLP |
| D4-RETMETID | D4-RETMETID |
| D4-RETRIEVE | D4-RETRIEVE |
| D4-SETATMCFG | D4-SETATMCFG |

Note: The following apply to all of the above outbound message types:


Copyright © 2026, Oracle and/or its affiliates.

- **Processing Method**: Real-time
- **Message XSL**: D4-Request xsl
- **Response XSL**: D4-Response.xsl

## Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the NES must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

**Service Provider - NES**

- **Service Provider**: NES
- **Description**: NES
- **External Reference ID**: NES
- **External System**: D4-NES
- **Out Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID

## Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the NES service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

### Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

### Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.


Copyright © 2026, Oracle and/or its affiliates.

### UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

### Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the NES service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

| Command | Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|---|
| Device Commission | Device Registration | D4-SetATMConfiguration | Set ATM Configuration |
| Device Decommission | Device Removal | D4-DecommMeterRequest | Decommission Meter Request |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | D4-ReadBillingData | Read Billing Data |
| On-Demand Read (Interval) | On-Demand Read (Interval) | D4-ReadLoadProfile | Read Load Profile |
| Device Status Check | Device Status Check | D4-ReadLoadStatus | Read Load Status |
| Remote Connect | Remote Connect | D4-ConnectLoad | Connect Load |
| Remote Disconnect | Remote Disconnect | D4-DisconnectLoad | Disconnect Load |

### Other Processing Methods

This section outlines details for other processing methods used by the NES adapter.

| Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|
| NES Add Meter Request | D4-AddMeterRequest | Commission |
| NES Retrieve Meter Identifier | D4-RetrieveMeterIdentifier | Commission Retrieve Meter Identifier |
| NES Retrieve Meter Information | D4-RetrieveMeterInfo | Retrieve Meter Info |

## Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation


Copyright © 2026, Oracle and/or its affiliates.

menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The NES endpoint override DVM (D4–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the NES adapter and the command used with each:

| DVM Key | Commands |
|---|---|
| DeviceManager | Remote Connect<br>Remote Disconnect<br>Device Commissioning<br>Device Decommissioning<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| EventManager | On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| GatewayManager | Remote Connect<br>Remote Disconnect<br>Device Commissioning<br>Device Decommissioning<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| SettingManager | Remote Connect<br>Remote Disconnect<br>Device Commissioning<br>Device Decommissioning<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| UserManager | Remote Connect<br>Remote Disconnect<br>Device Commissioning<br>Device Decommissioning<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| ProcessCallout | User Exit Functions |

To define an override Endpoint URI for the NES adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D4–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.


Copyright © 2026, Oracle and/or its affiliates.

- In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon ("+") to add a new key.

   The **Add Domain Values** dialog opens.
5. Enter the appropriate values in the **Add Domain Values** dialog as follows:
   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.
6. Click **OK**.

   The new DVM value will appear in the list of keys.
7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

   Click **Discard** to discard your changes.

   Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

## Configuring Networked Energy Services Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Networked Energy Services.

This section outlines some of the extendable lookups that must be configured for use with the NES adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

### Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the "Standard Event Name" extendable lookup.


Copyright © 2026, Oracle and/or its affiliates.

### UOM Code to Standard UOM Mapping

Usage received from a utility's head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.

### Interval Status Code to Condition Mapping

Interval usage received from the NES can include NES interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The NES Interval Status Code to Condition Mapping extendable lookup is used to determine how to map NES interval status codes to standard status codes when receiving usage from the NES.

Each value defined for the NES Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The NES interval status code
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.
- **Description**: A description of the interval status code.
- **Priority**: The priority of the interval status code, represented as a numeric value where larger numbers correlate to higher priorities.

### Other Extendable Lookups

| Business Object | Description |
|---|---|
| D4-EntityTypeLookup | Entity Type |
| D4-ExtServiceReturnCodeLookup | External Service Return Code |
| D4-GatewayTypeLookup | Gateway Type |
| D4-IDTypeLookup | ID Type |
| D4-InformationReturnTypeLookup | Information Return Type |
| D4-LoadVoltageStatusTypeLookup | Load Voltage Status Type |
| D4-TaskPriorityLookup | Task Priority |
| D4-TimeZoneManagerLookup | Time Zone Manager |

## Configuring NES Usage and Event Extract Processing

Usage and events received from the NES server must be requested from Oracle Utilities Smart Grid Gateway. Extract requests for usage and events are triggered by a batch process, and sent to the NES via middleware. The NES compiles a batch of usage or events and sends it to BPEL services to create a file to be processed by Oracle Service Bus (OSB) services, it is then loaded as SGG events and/or usage.


Copyright © 2026, Oracle and/or its affiliates.

This section describes the processing performed by the extract request process, and the configuration steps involved.

- Extract Request Processing
- Extract Request Configuration Steps
- Executing Extract Requests

## Extract Request Processing

This section provides a detailed description of the extract request process.

- The “Usage / Event Extract Scheduler Monitor” batch process (D1-EXTSC) checks for “NES Usage Extract Scheduler” or “NES Event Extract Scheduler” activities that are in the “Active” state, and transitions these to the “Send Request” state. It also updates the “Latest Request Start Date/Time” and “Latest Request End Date/Time” fields on the activity based on the date and time of the request.
- When the activity enters the “Send Request” state, an Enter algorithm (either “Usage Extract Scheduler Send Request” (D4-USGSNDREQ) or “Event Extract Scheduler Send Request” (D4-EVTSNDREQ)) sends a request via an message sender to the middleware (via the GrabBatchUsage or GrabBatchEvent BPEL services), which in turn send the request to the NES server.
- After the request is sent, the “NES Usage Extract Scheduler” or “NES Event Extract Scheduler” activity to set back to the “Active” state.
- The NES server calls the BPEL services “GrabBatchEvents” and “GrabBatchUsage.” BPEL then routes the incoming events or usage to a file system, which is used by OSB components to process the usage or events.

## Extract Request Configuration Steps

This section outlines the objects that must be configured to support batch extracts of usage and events from an NES head-end system. To configure the NES Adapter to support sending extract requests for usage and events, you must do the following:

- Create message senders (one for usage, one for events) to send extract requests.
- Create Outbound Message Types (one for usage, one for events) for extract requests.
- Associate the message senders and Outbound Message Types to the External System used with the head-end system service provider
- Create “NES Usage Extract Scheduler Type” and “NES Event Extract Scheduler Type” activity types.
- Create “NES Usage Extract Scheduler” and “NES Event Extract Scheduler” activities.

The following sections provide details for configuring these object. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders, Outbound Message Types, and External Systems.

**Note**: Configuration of the OSB and BPEL services used by this process is done when installing and configuring the OSB and BPEL components of the NES Adapter.

#### Message Sender - NES Handle Batch Event

Create an message sender for event extracts as follows:

**Main Tab**:


Copyright © 2026, Oracle and/or its affiliates.

- **Message Sender**: Enter a code for the batch event message sender (D4-HNLBATEVT)
- **Description**: NES Handle Batch Event
- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes messages via HTTP real-time)
- **Active**: True (checked)
- **MSG Encoding**: UTF-8 message encoding

**Context Tab**:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/NES/GrabBatchEvents
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/NES/HandleBatchEvent/BatchEventHandler

  where:

  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed

#### Message Sender - NES Handle Batch Usage

Create an message sender for usage extracts as follows:

**Main Tab**:

- **Message Sender**: Enter a code for the batch usage message sender (D4-HNLBATUSG)
- **Description**: NES Handle Batch Usage
- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes messages via HTTP real-time)
- **Active**: True (checked)
- **MSG Encoding**: UTF-8 message encoding

**Context Tab**:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/NES/GrabBatchUsage
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/NES/HandleBatchUsage/BatchUsageHandler

  where:

  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager


Copyright © 2026, Oracle and/or its affiliates.

- **<EM_SERVER>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
- **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed

#### Outbound Message Type - NES Handle Batch Event

Create an Outbound Message Type for event extract requests as follows:

- **Outbound Message Type**: Enter a code for the batch event Outbound Message Type (D4-HNLBATEVT)
- **Description**: NES Handle Batch Event
- **Business Object**: D1-OutboundMessage
- **Priority**: - Priority 20 (or lower)

#### Outbound Message Type - NES Handle Batch Usage

Create an Outbound Message Type for usage extract requests as follows:

- **Outbound Message Type**: Select a code for batch usage Outbound Message Type (D4-HNLBATUSG)
- **Description**: NES Handle Batch Usage
- **Business Object**: D1-OutboundMessage
- **Priority**: Priority 20 (or lower)

#### External Systems

Add the event and usage extract Outbound Message Types and message senders to the external system used for the NES head-end system as follows:

- **External System**: NES
- **Description**: NES
- **Outbound Message Types**:

| Outbound Message Type | Message Sender |
|---|---|
| D4-HNLBATEVT | D4-HNLBATEVT |
| D4-HNLBATUSG | D4-HNLBATUSG |

Note: The following apply to the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D4-Request xsl
- **Response XSL**: D4-Response.xsl

#### Activity Type - NES Event Extract Scheduler

Create an Activity Type for event extract requests as follows:

- **Activity Type**: Enter a code for the activity type (NES-EVENT-EXTRACT)
- **Description**: Enter a description for the activity type (NES Event Extract Scheduler Type)
- **Related Transaction BO**: NES Event Extract Scheduler
- **Activity Type Status**: Active


Copyright © 2026, Oracle and/or its affiliates.

- **Outbound Message Type**: NES Handle Batch Event
- **External System**: The external system used for the NES head-end system
- **Exception Handling**:
  - **To Do Type**: Activity To Do Type
  - **To Do Role**: System Default Role
  - **Retry Frequency**: Select retry frequency in case if Activity reaches an error state (00:03:00)
  - **Maximum Retries**: Select max number of times the activity will be retried - since it reached the error state (1)
  - **Discard Reason**: Select from the dropdown if available.

#### Activity Type - NES Usage Extract Scheduler

Create an Activity Type for usage extract requests as follows:

- **Activity Type**: Enter a code for the activity type (NES-USAGE-EXTRACT)
- **Description**: Enter a description for the activity type (NES Usage Extract Scheduler Type)
- **Related Transaction BO**: NES Usage Extract Scheduler
- **Activity Type Status**: Active
- **Outbound Message Type**: NES Handle Batch Usage
- **External System**: The external system used for the NES head-end system
- **Exception Handling**:
  - **To Do Type**: Activity To Do Type
  - **To Do Role**: System Default Role
  - **Retry Frequency**: Select retry frequency in case if Activity reaches an error state (00:03:00)
  - **Maximum Retries**: Select max number of times the activity will be retried - since it reached the error state (1)
  - **Discard Reason**: Select from the dropdown if available.

#### Activity - NES Event Extract Scheduler

Initiate an Activity for event extract requests as follows:

1. Select **Main**->**Communication**->**Activity+**
2. Select "NES Event Extract Scheduler Type" from the **Activity Type** drop-down list.
3. Enter the number of hours for each extract request in the **Number of Hours of Data Request** field. The default is 24.
4. *Optional*: Enter start and end dates and times for the first extract request in the **Current Request Start Date/Time** and **Current Request End Date/Time** fields.

   Note that these are optional fields. If not populated, the values are populated as follows:

   - **Current Request Start Date/Time**: taken from the last request stop date/time (which is populated by the “Event Extract Scheduler Send Request” algorithm (D4-EVTSNDREQ)


Copyright © 2026, Oracle and/or its affiliates.

- **Current Request End Date/Time**: the Current Request Start Date/Time plus the number of hours in the **Number of Hours of Data Request** field.

**Activity - NES Usage Extract Scheduler**

Initiate an Activity for usage extract requests as follows:

1. Select **Main**->**Communication**->**Activity+**
2. Select "NES Usage Extract Scheduler Type" from the **Activity Type** drop-down list.
3. Enter the number of hours for each extract request in the **Number of Hours of Data Request** field. The default is 24.
4. *Optional*: Enter start and end dates and times for the first extract request in the **Current Request Start Date/Time** and **Current Request End Date/Time** fields.

   Note that these are optional fields. If not populated, the values are populated as follows:

   - **Current Request Start Date/Time**: taken from the last request stop date/time (which is populated by the “Usage Extract Scheduler Send Request” algorithm (D4-USGSNDREQ)
   - **Current Request End Date/Time**: the Current Request Start Date/Time plus the number of hours in the **Number of Hours of Data Request** field.

### Executing Extract Requests

To schedule and execute usage and device event extract requests, set up the “Usage / Event Extract Scheduler Monitor” batch process (D1-EXTSC) to run at a frequency appropriate to when you wish to retrieve usage and device events from the NES. For example, to retrieve usage and device events on a daily basis, schedule the “Usage / Event Extract Scheduler Monitor” batch process (D1-EXTSC) to run once every 24 hours.

## Sensus

The Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI supports communication with the Sensus Regional Network Interface (RNI), including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, status check, and on-demand read. The following table describes the attributes of the adapter:

| Attribute | Details |
|---|---|
| Currently Supported Version | Sensus RNI 4.4, 4.5.1, 4.10, 4.12, 4.14, 4.15, 4.16 |
| Protocol | MultiSpeak 4.1 |
| Market(s) | Worldwide |
| Architecture | Long range radio WAN (mesh) |

- Sensus Adapter Processing
- Configuring a Sensus Head-End System
- Configuring Sensus Extendable Lookups
- Using the Sensus Test Harness

### Sensus Adapter Processing


Copyright © 2026, Oracle and/or its affiliates.

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

## Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements
- Device Events
- Base Package Business Objects

### Initial Measurements

The usage data exported from the AMI head-end system as a file in Sensus RNI format is loaded into Oracle Utilities as Initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D6-USAGE-BASE** contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This can be upgraded without affecting the customization and environment settings added to SGG-D6-USAGE-CM.
2. **SGG-D6-USAGE-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D6-USAGE-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRawIMD | Determines if the initial measurement data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElementInterval | Holds the name of inbound web service for the interval IMD seeder. | |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| destinationRootElementScalar | Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval. | |
| dateTimeInUTC | Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. | true<br>false |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterUsage | Determines if usage should be filtered. | true<br>false |
| useExternalTOU | Indicates whether or not an externally reference TOU period should be used when processing measurement data. | true<br>false |
| scalarExpanding | Defines the number of register reads to process in scalar payloads. When defined, Sensus payload processing uses this setting to ensure that all the scalar reads included in a payload will be processed.<br>Default value is 2. | |

### Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D6-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

### Configuring Initial Measurement Publishing Output


Copyright © 2026, Oracle and/or its affiliates.

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |


Copyright © 2026, Oracle and/or its affiliates.

### Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **Sensus UOM Code to Standard UOM Mapping** extendable lookup (D6-HeadendUOMLookup) are passed into the system for processing.

### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

### Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

## Device Events

The device event data exported from the head-end system as a file in Sensus RNI format is loaded into Oracle Utilities as a Device Event. One of your configuration tasks is to customize the device events processing.

The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D6-EVENT-BASE** contains components responsible for “actual” processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D6-EVENT-CM project.
2. **SGG-D6-EVENT-CM** allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D6-EVENT-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the event data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElement | Holds the name of inbound web service for the device event seeder. | |
| dateTimeInUTC | Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. | true<br>false |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |

### Publishing Events

The Sensus adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

#### Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D6-EVENT-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

#### Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |

### Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **Sensus Device Event Mapping** extendable lookup (D6-DeviceEventMappingLookup) are passed into the system for processing.


Copyright © 2026, Oracle and/or its affiliates.

### Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D6-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

Base Package Business Objects


Copyright © 2026, Oracle and/or its affiliates.

The Sensus RNI adapter base package includes the following device and initial measurement business objects:

| Business Object Name | Description |
|---|---|
| D6-InitialLoadIMDInterval | Sensus Initial Load IMD - Interval<br>Used when loading Sensus interval measurements into the system for the first time. |
| D6-InitialLoadIMDScalar | Sensus Initial Load IMD - Scalar |
| D6-SmartMeter | Sensus Smart Meter |

## Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound / Outbound Service Configuration
- BPEL Processes

### Communication Flows

The table below lists the communications created for each Sensus RNI command:

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Remote Connect | Sensus – Initiate Connect Disconnect | Sensus – Connect / Disconnect State Change | Connect Device Completion Event |
| Remote Disconnect | Sensus – Initiate Connect Disconnect | Sensus – Connect / Disconnect State Change | Disconnect Device Completion Event |
| Device Commissioning | Sensus – Meter Add Notification | | Device Commissioning Completion Event |
| Device Decommissioning | Sensus – Meter Remove Notification | | Device Decommissioning Completion Event |
| On-Demand Read (Scalar) | Sensus – Initiate Meter Read By Meter ID | Sensus – Reading Changed Notification | Create IMD Completion Event |
| On-Demand Read (Interval) | Interval data not supported | | |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Status Check | Sensus – Initiate Outage Detection | Sensus – Outage Detection Event Notification | |

### Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D6-CCOMPEV (Create IMD Completion Event) algorithm on the Create Completion Events status of the D6-ReadingChgNotification (Sensus - Reading Changed Notif SOA based) business object.

The D6-VALDVCFIL (Check if Device has any Filtering errors) algorithm on the Validate status of the D6-ReadingChgNotification (Sensus - Reading Changed Notif SOA based) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D6-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

## Device Communication Base Package Business Objects

The Sensus RNI Adapter base package includes the following communication business objects:

| Business Object Name | Description |
|---|---|
| D6-ConnectDisconStateChgNtf | Sensus - Connect/Disconnect State Change |
| D6-InitiateConnectDisconnect | Sensus - Initiate Connect Disconnect |
| D6-InitiateMeterByMeterId | Sensus - Initiate Meter Read By Meter ID |
| D6-InitiateOutageDetection | Sensus - Initiate Outage Detection |
| D6-MeterAddNotification | Sensus - Meter Add Notification |
| D6-MeterRemoveNotifcation | Sensus - Meter Remove Notification |
| D6-OutageDetectEvtNotification | Sensus - Outage Detection Event Notification |
| D6-ReadingChgNotification | Sensus - Reading Changed Notification |
| D6-UnsolicitedEvtNotification | Sensus - Unsolicited Event Notification |

### Sensus Event Data Mapping

The Sensus event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

| Sensus Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Transaction ID (from Header record) | External Source Identifier | This is the file name. |
| Device Identifier | External Device Identifier | |
| Event Name | External Event Name | |


Copyright © 2026, Oracle and/or its affiliates.

| Sensus Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Event Creation Date/Time | Event Date/Time | |
| Device Type | External Device Type | This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form):<br>Meter<br>Collector<br>Router |
| Service Location ID | External Service Location ID | |
| Communication Module Serial Number | External Communication Module Identifier | |
| Event Category ID | External Event Category | |
| Event Severity | External Event Severity | Valid values include (although the element itself is free-form):<br>Alert<br>Information |
| Status Value | External Status Value | This represents additional information that relates to the event itself. |
| Status Date/Time | External Status Date/Time | The date & time at which the additional information referenced above had occurred. |

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgement and response messages are sent and received validating that commands have been transmitted.

| Outbound Message Type | Description |
|---|---|
| D6-CONDISCON | Sensus Initiate Connect Disconnect |
| D6-INITMTR | Initiate Meter Read By Meter ID |
| D6-INTOUTDET | Initiate Outage Detection Request |

Inbound / Outbound Service Configuration


Copyright © 2026, Oracle and/or its affiliates.

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Sensus RNI includes the following inbound web services:

| Inbound Web Service | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Used for upload of device events.<br>The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event.<br>If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created. |
| D1-InitialLoadIMD | Used for initial measurement upload.<br>The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system. |
| D1-DeviceStatusCheck | Device Status Check<br>This service is invoked by the integration layer to instantiate a Device Status Check command. |
| D1-InitialLoadIMD | Used by OSB to instantiate an IMD<br>This inbound web service is used by OSB to instantiate an Initial Measurement Data for incoming interval usage in the Sensus format. |
| D1-RemoteConnect | Remote Connect<br>This service is invoked by the integration layer to instantiate a Remote Connect command. |
| D1-RemoteDisconnect | Remote Disconnect<br>This service is invoked by the integration layer to instantiate a Remote Disconnect command. |
| D6-ConDisconStChgNotification | Initiate Connect Disconnect response.<br>Retrieve response from the Initiate Connect Disconnect command. |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service | Description |
|---|---|
| D6-OutageDetectionEventNotification | Initiate Outage Detection Response<br>Retrieve response from the Initiate Outage Detection Event Notification command. |
| D6-ReadingChangedNotification | Reading Changed Notification<br>Notification that a Sensus device reading has changed. |
| D6-UnsolicitedEventNotification | Unsolicited Event Response<br>Retrieve unsolicited notifications when an event triggers an alarm on the meter. |

### Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway adapter for Sensus RNI includes the following message senders:

| Message Sender | Description |
|---|---|
| D6-CONDISCON | Sensus Initiate Connect/Disconnect |
| D6-INTOUTDET | Initiate Outage Detection Request |
| D6-InitMID | Initiate Meter Read By Meter ID Outbound Message |

## BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to MultiSpeak 4.1 format, invoking process callouts and invoking the remote endpoint to trigger the device events.

**OnDemandRead Composite Process** — Invokes the remote endpoint to trigger the on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

**ConnectDisconnect Composite Process** — Invokes the remote endpoint to trigger the connect/disconnect event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

**CommissionDecommission Composite Process** — Invokes the remote endpoint to trigger the commission or decommission event. After the synchronous call completes, a one of the following second business callout services is invoked to determined if the related “received” or “completed” callout should be executed:

- isExecutingCommissionReceivedCallout
- isExecutingCommissionCompletedCallout
- isExecutingDecommissionReceivedCallout
- isExecutingDecommissionCompletedCallout

**DeviceStatusCheck Composite** — Invokes the remote endpoint to trigger the initiate outage detection event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.


Copyright © 2026, Oracle and/or its affiliates.

**ProcessCallout Composite** — This business callout provides a point at which customers and implementers can incorporate custom business logic and transformations. This composite includes the WSDLs and processing logic for all of the MultiSpeak processes. The default implementation of each method is a direct return of the input.

### Web Services

These web services are all defined in the Sensus RNI head end system. The WSDLs were added to a Meta Data Storage (MDS) layer in OUAF and all references to the WSDL point to this MDS location. These web services have HTTP security by default. You may need to modify the security as a part of your implementation.

| Web Service | Related BPEL Process | Description |
|---|---|---|
| CB_ServerService | ConnectDisconnect | This web service defines the return interface, the means by which the status is returned to the calling system.<br>This web service is only be invoked by the head end system, not OUAF. Only the CDStateChangeNotification web method is implemented in the composite.<br>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/ConnectDisconnect/CB_ServerService |
| CB_Server | OnDemandRead | This web service defines the return interface, the means by which the reading is returned to the calling system.<br>This web service is only be invoked by the head end system, not OUAF. Only the ReadingChangedNotification web method is implemented in the composite.<br>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/OnDemandRead/CB_Server |
| OA_ServerService | DeviceStatusCheck | This web service defines the asynchronous return for InitateOutageDetectionEventRequest for solicited responses. It is also used for unsolicited alarms.<br>This web service is only be invoked by the head end system, not OUAF. Only the ODEventNotification, PingURL, and GetMethods web methods are implemented in the composite.<br>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/DeviceStatusCheck/OA_ServerService |

### Sensus RNI Web Services

The following table describes the Sensus RNI web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:


Copyright © 2026, Oracle and/or its affiliates.

| Smart Grid Gateway Command | AMI Adapter Business Objects | Sensus Web Services | Sensus Operations |
|---|---|---|---|
| Device Commissioning | D6-MeterAddNotification | MR | MeterAddNotification |
| Device Decommissioning | D6-MeterRemoveNotification | MR | MeterRemoveNotification |
| Remote Connect/ Remote Disconnect | D6-InitiateConnectDisconnect | CD | InitiateConnectDisconnect |
| | D6-ConnectDisconStateChgNtf | CB | CDStatesChangedNotification (async reply) |
| Device Status Check | D6-InitiateOutageDetection | OD | InitiateOutageDetectionEventRequest |
| | D6-OutageDetectEvtNotification | OA | ODEventNotification (async reply) |
| On-Demand Read | D6-InitiateMeterByMeterId | MR | InitiateMeterReadingsByMeterID |
| | D6-ReadingChangedNotification | CB | ReadingChangedNotification (async reply) |

## Configuring a Sensus Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI to communicate with the Sensus RNI.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the Sensus Regional Network Interface (RNI) in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

| Inbound Web Service Name | Description |
|---|---|
| D1-DeviceEventSeeder | Device Event Seeder |
| D1-DeviceStatusCheck | Device Status Check |
| D1-InitialLoadIMD | IMD Seeder |
| D1-PayloadErrorNotif | Payload Error Notification |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service Name | Description |
|---|---|
| D1-PayloadStatistics | Payload Statistics |
| D1-PayloadSummary | Payload Summary |
| D6-ConDisconStChgNotification | Initiate Connect Disconnect Response |
| D6-OutageDetectionEventNotification | Initiate Outage Detection Response |
| D6-ReadingChangedNotification | Reading Changed Notification |
| D6-UnsolicitedEventNotification | Unsolicited Event Response |

Note: The following apply to all of the above inbound web services:

**Message Options**

- **Trace**: No
- **Debug**: No
- **Active**: Yes

**Operations**

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

## Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

| Message Sender | Description |
|---|---|
| D6-CONDISCON | Sensus Initiate Connect/Disconnect |
| D6-INTOUTDET | Initiate Outage Detection Request |
| D6-InitMID | Initiate Meter Read By Meter ID Outbound Message |

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**


Copyright © 2026, Oracle and/or its affiliates.

- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/Sensus/**<SERVICE>**
- where:
  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
  - **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

## Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

| Outbound Message Type | Description |
|---|---|
| D6-CONDISCON | Sensus Initiate Connect Disconnect |
| D6-INITMTR | Initiate Meter Read By Meter ID |
| D6-INTOUTDET | Initiate Outage Detection Request |

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

## External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the Sensus RNI must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

**External System - Sensus**:

- **External System**: Sensus
- **Description**: Sensus


Copyright © 2026, Oracle and/or its affiliates.

- **Outbound Message Types**:

| Outbound Message Type | Description | Message Sender |
|---|---|---|
| D6-CONDISCON | Sensus Initiate Connect Disconnect | D6-CONDISCON |
| D6-INITMTR | Initiate Meter Read By Meter ID | D6-InitMID |
| D6-INTOUTDET | Initiate Outage Detection Request | D6- INTOUTDET |

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D6-Request xsl
- **Response XSL**: D6-Response.xsl

## Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the Sensus RNI must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about creating service providers.

**Service Provider - Sensus**:

- **Service Provider**: Sensus
- **Description**: Sensus
- **External Reference ID**: Sensus
- **External System**: Sensus
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID

## Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the Sensus service provider. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about configuring processing methods.


Copyright © 2026, Oracle and/or its affiliates.

### Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

### Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

### UOM Translation

UOM mapping processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

### Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the Sensus service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

| Command | Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|---|
| Device Commission | Device Registration | D6-MeterAddNotification | Commission |
| Device Decommission | Device Removal | D6-MeterRemoveNotification | Decommission |
| Device Status Check | Device Status Check | D6-InitiateOutageDetection | Initiate Outage Detection Request |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | D6-InitiateMeterByMeterId | Initiate Meter Read by Meter ID |
| Remote Connect | Remote Connect | D6-InitiateConnectDisconnect | Connect Device |
| Remote Disconnect | Remote Disconnect | D6-InitiateConnectDisconnect | Disconnect Device |

## Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation


Copyright © 2026, Oracle and/or its affiliates.

menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an "Endpoint Override" Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Sensus endpoint override DVM (D6–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Sensus adapter and the command used with each:

| DVM Key | Commands |
|---|---|
| MR_Server | Device Commissioning<br>Device Decommissioning<br>On-Demand Read (Scalar) |
| CD_Server | Remote Connect<br>Remote Disconnect |
| OD_Server | Device Status Check |
| ProcessCallout | User Exit Functions |

To define an override Endpoint URI for the Sensus adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D6–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon ("+") to add a new key.

   The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.

6. Click **OK**.

   The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.


Copyright © 2026, Oracle and/or its affiliates.

Changes take effect immediately upon publishing the session.

Click **Discard** to discard your changes.

Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

## Configuring Sensus Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Sensus.

This section outlines some of the extendable lookups that must be configured for use with the Sensus adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

### Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

### UOM Code to Standard UOM Mapping

Usage received from a utility’s head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-end UOM**: The unit of measure code used by the head-end system.
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive)

### Interval Status Code to Condition Mapping

Interval usage received from the Sensus RNI can include Sensus interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Sensus Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Sensus interval status codes to standard status codes when receiving usage from the Sensus RNI.

Each value defined for the Sensus Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The Sensus interval status code


Copyright © 2026, Oracle and/or its affiliates.

- **Description**: A description of the interval status code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

**Other Extendable Lookups**

| Business Object Name | Description |
|---|---|
| D6-CDReasonCodeLookup | Sensus Connect/Disconnect Reason Code |
| D6-LoadActionCodeLookup | Sensus Load Action Code |
| D6-OutageEventTypeLookup | Sensus Outage Event Type |
| D6-SensusTimeUnits | Sensus Time Units |
| D6-ServiceTypeMappingLookup | Sensus Service Type Mapping |

## Using the Sensus Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI includes a test harness that can be configured to simulate a general head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Test Harnesses for more information.

# Silver Spring Networks

The Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks supports communication with the Silver Spring Networks UtilityIQ application, including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, status check, and on-demand read. The following table describes the attributes of the adapter:

| Attribute | Details |
|---|---|
| Currently Supported Versions | UtilityIQ Version 4.4, 4.7, 4.10, 4.14, 5.0, 5.1, 5.2, 6.1 |
| Protocol | Proprietary<br>MultiSpeak 4.1 |
| Market(s) | Worldwide |
| Architecture | RF WAN (mesh) based on Access Points |

- Silver Spring Networks Adapter Processing
- Configuring a Silver Spring Networks Head-End System
- Configuring Silver Spring Networks Extendable Lookups
- Using the Silver Spring Networks Test Harness

## Silver Spring Networks Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and Oracle Utilities Application Framework (OUAF) objects that are supplied as part of the base package.


Copyright © 2026, Oracle and/or its affiliates.

This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.

- Initial Measurement Data and Device Event Loading
- Device Communication

## Initial Measurement Data and Device Event Loading

The initial measurement data load and device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway (SGG) service format. Payloads contain measurements and meter events in some head-end specific format. OSB then places each service call into a Java Message Service (JMS) queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel. A service then creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements
- Device Events
- Base Package Business Objects

### Initial Measurements

The usage data exported from the AMI head-end system as a file in Silver Spring Networks XML format is loaded into Oracle Utilities as initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. SGG-D7-SSNXML-BASE contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to SGG-D7-SSNXML-CM.
2. SGG-D7-SSNXML-CM allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D7-SSNXML-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the initial measurement data is populated as raw data. Valid values are: | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. Valid values are: | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. Valid values are: | true<br>false |
| destinationRootElementInterval | Holds the name of inbound web service for the interval IMD seeder. | |
| destinationRootElementScalar | Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval. | |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| destinationRootElementEvent | Holds the name of inbound web service for the device event seeder. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |
| filterUsage | Determines if usage should be filtered. | true<br>false |

### Publishing Initial Measurement Data

The SGG adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

#### Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D7-SSNXML-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

#### Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published through this feature is published prior to validation, estimation, and editing (VEE) processing. In addition, filtering *cannot* be applied to data published through this feature.


Copyright © 2026, Oracle and/or its affiliates.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |

### Filtering Initial Measurement Data

The SGG adapter can be configured to filter initial measurement data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **SSN - UOM Code to Standard UOM Mapping** extendable lookup (D7-HeadendUOMLookup) are passed into the system for processing.

### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This


Copyright © 2026, Oracle and/or its affiliates.

enables large files to be completely copied to the input directory before they are retrieved for processing.

### Prioritized Initial Measurement Processing

The SGG adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition, initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

## Device Events

The device event data exported from the head-end system as a file in Silver Spring Networks CSV format is loaded into Oracle Utilities as a device event. One of your configuration tasks is to customize the device events processing.

The required functionality is delivered in the base product as two OSB projects:

1. SGG-D7-CSV-BASE contains components responsible for “actual” processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D7-CSV-CM project.
2. SGG-D7-CSV-CM allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D7-CSV-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

| Element | Description | Valie Values |
|---|---|---|
| populateRaw | Determines if the device event data is populated as raw data. Valid values are: | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. Valid values are: | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. Valid values are: | true<br>false |
| destinationRootElement | Holds the name of inbound web service for the device event seeder. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. Valid values are: | true<br>false |

### Publishing Events

The Silver Spring Networks adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Device Event Publishing


Copyright © 2026, Oracle and/or its affiliates.

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D7-CSV-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

**Configuring Device Event Publishing Output**

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems. Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Description | Default Value |
|---|---|---|
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |

### Filtering Events

The SGG adapter can be configured to filter event data passed into SGG and the Oracle Utilities Meter Data Management application. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **SSN - Device Event Mapping** extendable lookup (D7-DeviceEventMappingLookup) are passed into the system for processing.

### Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)


Copyright © 2026, Oracle and/or its affiliates.

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D7-CSV-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.
8. Enter "false" in the **value** column.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

### Base Package Business Objects

The Silver Spring Networks adapter base package includes the following device and initial measurement business objects:

| Business Object Name | Description |
|---|---|
| D7-InitialLoadIMDInterval | SSN - Initial Load IMD - Interval<br>Used when loading Silver Spring Network (SSN) interval measurements into the system for the first time. |
| D7-InitialLoadIMDScalar | SSN - Initial Load IMD - Scalar<br>Used when loading Silver Spring Network (SSN) scalar measurements into the system for the first time. |
| D7-SmartMeter | SSN — Smart Meter |

## Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related web service from the head-end system. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.


Copyright © 2026, Oracle and/or its affiliates.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Outbound Message Types
- Inbound/Outbound Service Configuration
- BPEL Processes

Communication Flows

The table below lists the communications created for each Silver Spring Networks command:

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Remote Connect | SSN–Connect or Disconnect | SSN – Connect or Disconnect Response<br>Remote Provisioning Job (Get Status) | Connect Device Completion Event |
| Remote Disconnect | SSN–Connect or Disconnect | SSN – Connect or Disconnect Response | Disconnect Device Completion Event |
| Device Commissioning (see Note below) | SSN- Replace Location | | Device Commissioning Completion Event |
| Device Decommissioning | SSN- Replace Device At Location (Decomm) | | Device Decommissioning Completion Event |
| On-Demand Read (Scalar) | SSN – Add Meter Read Job (Scalar) | SSN – Meter Read Response (Scalar) | Create IMD Completion Event |
| On-Demand Read (Interval) | SSN – Add Meter Read Job (Interval) | SSN – Meter Read Response (Interval) | Create IMD Completion Event |
| Device Status Check | SSN – Add Ping job | SSN – Ping Job Response | |

> **Note**
>
> Service points must be provisioned in the Silver Spring Networks (Itron AMM) system before the Device Commissioning command can be used.

**Unit of Measure Filtering for On-Demand Reads**

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D7-CCOMPEV (Create Completion Event Scalar) algorithm on the Create Completion Events status of the D7-MeterReadResponseScalar (SSN - Meter Read Response (Scalar)) business object (for scalar on-demand read commands), and/or on the D7-CCOMPEVIN (Create Completion Event Interval) algorithm on the Create Completion Events status of the D7-MeterReadResponseInterval (SSN - Meter Read Response (Interval)) business object (for interval on-demand read commands).


Copyright © 2026, Oracle and/or its affiliates.

When filtering is enabled, the “Filter if invalid external MC identifier” parameter on the D7-CCOMPEV and D7-CCOMPEVIN algorithms controls what should happen if a measuring component cannot be found using the MC Identifier Number sent from the head end system. When set to 'Yes' and no measuring component is found, the initial measurement will be filtered. When set to 'No' and no measuring component is found, the system will attempt to find the measuring component using the UOM and TOU. The value of the “Filter if invalid external MC identifier” on these algorithms must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either "yes" or "no".

The D7-VALDVCFLS (SSN - Check if Device has any Filtering errors (Scalar)) algorithm on the Validate status of the D7-MeterReadResponseScalar (SSN - Meter Read Response (Scalar)) business object is used to check for errors related to filtering for scalar on demand read commands. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D7-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled). The value of the “Filter if invalid external MC identifier” on this algorithm must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either "yes" or "no".

The D7-VALDVCFLI (SSN - Check if Device has any Filtering errors (Interval)) algorithm on the Validate status of the D7-MeterReadResponseInterval (SSN - Meter Read Response (Interval)) business object is used to check for errors related to filtering for interval on demand read commands. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D7-CCOMPEVIN algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled). The value of the “Filter if invalid external MC identifier” on this algorithm must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either "yes" or "no".

#### Receiving Results via JMS Queue

Some Silver Spring Networks commands can be configured to return results to a JMS queue, eliminating the need for the system to periodically poll for completed commands. This feature is enabled by setting the **Receive Job Results via JMS** parameter to "Yes" on the communication type used by the command.

The table below lists the commands that support this feature and the communication type business objects that correspond to each.

| Command | Communication Type Business Object |
|---|---|
| Connect/Disconnect | D7-ConnectDisconnectType |
| Device Status Check | D7-AddPingJobType |
| On-Demand Read (Interval) | D7-AddMeterReadJobIntervalType |
| On-Demand Read (Scalar) | D7-AddMeterReadJobScalarType |

The JMS queue to which command results are sent (all commands send results to the same JMS queue) is defined during installation. Refer to **Deploying the SOA Adapter for Silver Spring Networks** in the *Oracle Utilities Smart Grid Gateway Installation Guide* for details about creating JMS queues.

### Device Communication Base Package Business Objects

The Silver Spring Networks Adapter base package includes the following communication business objects:


Copyright © 2026, Oracle and/or its affiliates.

| Business Object Name | Description |
|---|---|
| D7-AddMeterReadJobInterval | SSN - Add Meter Read Job (Interval) |
| D7-AddMeterReadJobScalar | SSN - Add Meter Read Job (Scalar) |
| D7-AddPingJob | SSN - Add Ping Job |
| D7-ConnectDisconnect | SSN - Connect or Disconnect |
| D7-ConnectDisconnectResp | SSN - Connect or Disconnect Response |
| D7-GetStatus | SSN - Get Status |
| D7-GetStatusResponse | SSN - Get Status Response |
| D7-MeterReadResponseInterval | SSN - Meter Read Response (Interval) |
| D7-MeterReadResponseScalar | SSN - Meter Read Response (Scalar) |
| D7-PingJobResponse | SSN - Ping Job Response |
| D7-ReplaceDeviceAtLocForDecomm | SSN - Replace Device At Location (Decomm) |
| D7-ReplaceLocation | SSN - Replace Location |

### Silver Spring Networks Event Data Mapping

The Silver Spring event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

| Silver Spring Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Transaction ID (from Header record) | External Source Identifier | This is the file name. |
| Device Identifier | External Device Identifier | |
| Event Name | External Event Name | |
| Event Creation Date/Time | Event Date/Time | |
| Device Type | External Device Type | This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form):<br>Meter<br>Collector<br>Router |
| Service Location ID | External Service Location ID | |
| Communication Module Serial Number | External Communication Module Identifier | |
| Event Category ID | External Event Category | |
| Event Severity | External Event Severity | Valid values include (although the element itself is free-form):<br>Alert<br>Information |
| Status Value | External Status Value | This represents additional information that relates to the event itself. |


Copyright © 2026, Oracle and/or its affiliates.

| Silver Spring Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Status Date/Time | External Status Date/Time | The date & time at which the additional information referenced above had occurred. |

External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgment and response messages are sent and received validating that commands have been transmitted.

| Outbound Message Type | Description |
|---|---|
| D7-COMMS | Replace Device At Location |
| D7-OB MSG TY | Outbound Message Type SSN |

Inbound/Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Oracle Utilities Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

### Inbound web services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Silver Spring Networks includes the following inbound web services:

| Inbound Web Service | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service | Description |
|---|---|
| D1-DeviceEventSeeder | Used for upload of device events.<br>The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event.<br>If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created. |
| D1-DeviceStatusCheck | Device Status Check<br>This service is invoked by the integration layer to instantiate a Device Status Check command. |
| D1-InitialLoadIMD | Used by OSB to instantiate an IMD<br>The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system. |
| D1-RemoteConnect | Remote Connect<br>This service is invoked by the integration layer to instantiate a Remote Connect command. |
| D1-RemoteDisconnect | Remote Disconnect<br>This service is invoked by the integration layer to instantiate a Remote Disconnect command. |
| D7-ConnectDisconnectResponse | Connect Disconnect Response<br>Retrieves response for Remote provisioning Job Connect or Disconnect commands. |
| D7-GetStatusResponse | D7-GetStatusResponse<br>Retrieve response from the Get Status command. |
| D7-MeterReadResponseInterval | SSN - Meter Read Response (Interval) |
| D7-MeterReadResponseScalar | SSN - Meter Read Response (Scalar) |
| D7-PingJobResponse | SSN - Ping Response<br>Retrieves response from the Ping Job Response command. |

### Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

## BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to Silver Spring Networks format, invoking process callouts and invoking the remote endpoint to trigger the device events.

**OnDemandRead Composite Process** — Provides access points to edge application and handles data between edge application and head end system. It invokes sequence of web methods to head end system and retrieves meter read and send it back to Edge application.


Copyright © 2026, Oracle and/or its affiliates.

**ConnectDisconnect Composite Process** — Performs the conversion from Oracle Utilities format to SSN format, invokes process callouts, and invokes the remote endpoint to trigger the connect event. A second, asynchronous reply will call back into the OUAF layer when the status change is completed at the head-end system. Another asynchronous reply will call back into the OUAF layer to send Meter Read Results.

**CommissionDecommission Composite Process** — Performs the conversion from Oracle Utilities format to SSN UIQ format, invokes process callouts, and invokes the remote endpoint to trigger the commission or decommission of meter.

**DeviceStatusCheck Composite** — Performs the conversion from Oracle Utilities format to SSN format, invokes process callouts, and makes a call via a proxy to the head-end system starting the Meter Ping operation. In an ideal scenario, the job status is returned as completed and the results are acquired and sent back to OUAF. If the job takes longer, OUAF will initiate a second request that will poll the head end system for the job status. When the job is completed, the results are returned to OUAF.

**Common Composite** — Contains two main classes of operations: Proxies and ProcessCallouts. Proxies are simple mediators that forward a web service call to a preset endpoint. No transformations are performed. They are convenient because they allow head end URLs and security to be set in a single composite. ProcessCallouts are points of customization which allow users to modify data and/or initiate some external business process.

**BulkRequest Composite** — Provides access points to requesting application. It decouples the bulk request into single commands for each meter/device in the request and sends it to edge application for processing.

#### Web Services

The following web services are all defined in the Silver Spring Networks head-end system:

- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** ReplaceDeviceAtLocation
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/CommissionDecommission/CommissionDecommissionService
- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** ReplaceLocation
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/CommissionDecommission/CommissionDecommissionService
- **ConnectDisconnectService**
  - **BPEL Process:** ConnectDisconnect
  - **Operation:** AddRemoteProvisioningJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/ConnectDisconnect/ConnectDisconnectService
- **DeviceStatusCheckService**
  - **BPEL Process:** DeviceStatusCheck
  - **Operation:** AddPingJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/DeviceStatusCheck/DeviceStatusCheckService


Copyright © 2026, Oracle and/or its affiliates.

- **AddMeterReadJobService**
  - **BPEL Process:** AddMeterRead
  - **Operation:** AddMeterReadJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/OnDemandRead/AddMeterReadJobService
- **OnDemandReadService**
  - **BPEL Process:** OnDemandRead
  - **Operation:** GetJobStatus
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/OnDemandRead/OnDemandReadService

### Silver Spring Networks Utility IQ Web Services

The following table describes the Silver Spring Networks Utility IQ web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

| Smart Grid Gateway Command | AMI Adapter Business Objects | Silver Spring Networks Web Services | Silver Spring Networks Operations |
|---|---|---|---|
| Device Commissioning | D7-ReplaceLocation | Device Manager | findDevice<br>ReplaceLocation |
| Device Decommissioning | D7-ReplaceDeviceAtLocForDecomm | Device Manager | findDevice<br>ReplaceDeviceAtLocation |
| Remote Connect/ Remote Disconnect | D7-ConnectDisconnect<br>D7-GetStatus | Device Manager<br>Job Manager<br>DeviceResults | findDevice<br>addRemoteProvisioningJob<br>getJobStatusForDevice<br>getRemoteProvisioningResultsByJobID<br>findJob<br>getJobStatus<br>getMeterReadResultsByJobID |
| Device Status Check | D7-AddPingJob | Device Manager<br>Job Manager<br>DeviceResults | findDevice<br>addPingJob, getJobStatus<br>getPingResultsByJobID |
| On-Demand Read | D7-AddMeterReadJobInterval<br>D7-AddMeterReadJobScalar | Device Manager<br>Job Manager<br>DeviceResults | findDevice<br>addMeterReadJob<br>getJobStatus<br>getMeterReadResultsByJobID |

## Configuring a Silver Spring Networks Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks to communicate with the Silver Spring Networks UtilityIQ.

- Inbound Web Services


Copyright © 2026, Oracle and/or its affiliates.

- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

## Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the Silver Spring Networks UtilityIQ application in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

| Inbound Web Service Name | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Device Event Seeder |
| D1-DeviceStatusCheck | Device Status Check |
| D1-InitialLoadIMD | Used by OSB to instantiate an IMD |
| D1-PayloadErrorNotif | Payload Error Notification |
| D1-PayloadStatistics | Payload Statistics |
| D1-PayloadSummary | Payload Summary |
| D7-ConnectDisconnectResponse | Connect Disconnect Response |
| D7-GetStatusResponse | Get Status Response |
| D7-MeterReadResponseInterval | SSN - Meter Read Response (Interval) |
| D7-MeterReadResponseScalar | SSN - Meter Read Response (Scalar) |
| D7-PingJobResponse | SSN - Ping Response |

Note: The following apply to all of the above inbound web services:

**Message Options**

- **Trace**: No
- **Debug**: No
- **Active**: Yes

**Operations**

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add


Copyright © 2026, Oracle and/or its affiliates.

## Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests. An message sender should be configured for each command.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

| Message Sender | Description |
|---|---|
| D7-DECOMM | SSN Replace Device at Location for Decommission |
| D7-COMM | SSN Replace Location - Commission |
| D7-ADDJOB | SSN Add Meter Read Job |
| D7-ADDPING | SSN Add Ping Job |
| D7-CONNECT | SSN Connect Device |
| D7-GTSTATUS | SSN Get Status |

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/ssn/**<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/SSN/**<SERVICE>**
- where:
- **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
- **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
- **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
- **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
- **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
- **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

## Outbound Message Types


Copyright © 2026, Oracle and/or its affiliates.

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

| Outbound Message Type | Description |
|---|---|
| D7-COMMS | Replace Device At Location |
| D7-OB MSG TY | Outbound Message Type SSN |

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

## External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the Silver Spring Networks UtilityIQ must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

**External System - Silver Spring Networks**:

- **External System**: Silver Spring Networks
- **Description**: Silver Spring Networks
- **Outbound Message Types**:

| Outbound Message Type | Description |
|---|---|
| D7-OB MSG TY | Outbound Message Type SSN |

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: D7-Request xsl
- **Response XSL**: D7-Response.xsl

## Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.


Copyright © 2026, Oracle and/or its affiliates.

A service provider that represents the Silver Spring Networks UtilityIQ must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

**Service Provider - Silver Spring Networks**:

- **Service Provider**: Silver Spring Networks
- **Description**: Silver Spring Networks
- **External Reference ID**: Silver Spring Networks
- **External System**: Silver Spring Networks
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **Pre-Commissioning Device ID Type**:
- **AMI Measuring Component ID Type**: Channel ID

## Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including as bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to a given head-end system.

The following types of processing methods must be configured for the Silver Spring Networks service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.

### Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head-end system.

### Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

### Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the Silver Spring Networks service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|---|
| Device Commission | Device Registration | D7-ReplaceLocation | Commission |
| Device Decommission | Device Removal | D7-ReplaceDeviceAtLocForDecomm | Decommission |
| Device Status Check | Device Status Check | D7-AddPingJob | Get Status |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | D7-AddMeterReadJobScalar | Read Meter Data |
| On-Demand Read (Interval) | On-Demand Read (Interval) | D7-AddMeterReadJobInterval | Read Meter Data |
| Remote Connect | Remote Connect | D7-ConnectDisconnect | Connect Device |
| Remote Disconnect | Remote Disconnect | D7-ConnectDisconnect | Disconnect Device |

## Configuring Endpoint URIs

Part of the configuration process is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands. The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head-end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoint URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Silver Springs Network endpoint override DVM (D7–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Silver Springs Network adapter and the command used with each:

| DVM Key | Commands |
|---|---|
| DeviceResults4.4 | Remote Connect |
| DeviceResults4.7 | Remote Disconnect |
| DeviceResults4.10 | On-Demand Read (Scalar)<br>On-Demand Read (Interval)<br>Device Status Check |
| DataAggregation4.4<br>DataAggregation4.7<br>DataAggregation4.10 | Not used in this version. |


Copyright © 2026, Oracle and/or its affiliates.

| DVM Key | Commands |
|---|---|
| DeviceManager4.4<br>DeviceManager4.7<br>DeviceManager4.10 | Device Commissioning<br>Device Decommissioning<br>Remote Connect<br>Remote Disconnect<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval)<br>Device Status Check |
| JobManager4.4<br>JobManager4.7<br>JobManager4.10 | Remote Connect<br>Remote Disconnect<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval)<br>Device Status Check |
| ProcessCallout | User Exit Functions |

> **Note**
>
> The numbers in the keys above (4.4, 4.7, and 4.10) designate the version of the Silver Springs Network IQ head-end system.
>
> Only a single version of each key should be defined, based on the version of the Silver Springs Network IQ head-end system

To define an override Endpoint URI for the Silver Springs Network adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the D7–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon (“+”) to add a new key.

   The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.

6. Click **OK**.


Copyright © 2026, Oracle and/or its affiliates.

The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.

   Enter an optional note for the session in the **Publish Session** dialog.

   Changes take effect immediately upon publishing the session.

   Click **Discard** to discard your changes.

   Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

## Configuring Silver Spring Networks Extendable Lookups

> **Note**
>
> This section applies to **both** middleware and native implementations of the Smart Grid Gateway Adapter for Silver Spring Networks.

This section outlines some of the extendable lookups that must be configured for use with the Silver Spring Networks adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

### Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system. Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system.
- **Description**: A description of the device event.
- **Status**: The status of the lookup value (can be Active or Inactive).
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

### UOM Code to Standard UOM Mapping

Usage received from a utility’s head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The UOM Code to Standard UOM Mapping extendable lookup is used for this purpose. Each value defined for the UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-End Unit of Measure**: The unit of measure code used by the head-end system.
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive).
- **Unit of Measure**: The unit of measure defined in the system.

### Interval Status Code to Condition Mapping

Interval usage received from the Silver Spring Networks UtilityIQ can include Silver Spring Networks interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Silver


Copyright © 2026, Oracle and/or its affiliates.

Spring Networks Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Silver Spring Networks interval status codes to standard status codes when receiving usage from the Silver Spring Networks UtilityIQ.

Each value defined for the Silver Spring Networks Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The Silver Spring Networks interval status code
- **Description**: A description of the interval status code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

### Other Extendable Lookups

| Business Object Name | Description |
|---|---|
| D7-ActivationStatusLookup | SSN - Activation Status |
| D7-ChannelStatusLookup | SSN - Channel Status |
| D7-DataTimeZoneLookup | SSN - Data Time Zone |
| D7-DeviceStatusLookup | SSN - Device Status |
| D7-ExecutionStatusLookup | SSN - Execution Status |
| D7-HeadendSQILookup | SSN - Head-End SQI to Standard SQI |
| D7-HeadendTOULookup | SSN - Head-End TOU to Standard TOU |
| D7-IntervalStatusLookup | SSN - Interval Status |
| D7-MeterConnectionStatusLookup | SSN - Meter Connection Status |
| D7-MeterReadTypeLookup | SSN - Meter Read Type |
| D7-OldAdminStateNameLookup | SSN - Old Admin State Name |
| D7-PriorityLookup | SSN - Priority Status |
| D7-ProvisioningActionLookup | SSN - Provisioning Action |
| D7-ProvisioningCommandStatus | SSN - Provisioning Command Status |
| D7-ProvisioningDataType | SSN - Provisioning Data Type |
| D7-ReadModeLookup | SSN - Read Mode |
| D7-RegisterReadSourceLookup | SSN - Register Read Source |
| D7-RegisterStatusLookup | SSN - Register Status |
| D7-SSNVersionLookup | SSN - Version |
| D7-StatusFlagNameLookup | SSN - Status Flag Name |

## Using the Silver Spring Networks Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks (SSN) includes a test harness that can be configured to simulate the Silver Spring Networks UtilityIQ head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. See Using Smart Grid Gateway Test Harnesses for more information.

# Using Smart Grid Gateway Middleware Test Harnesses


Copyright © 2026, Oracle and/or its affiliates.

The Oracle Utilities Smart Grid Gateway Adapter includes middlware test harnesses for each supported head end system. The test harnesses can be configured to simulate a head-end system for initial testing of one-way or two-way commands. The test harnesses are Multispeak 3.0 standard compliant and include a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. This section describes the general components of the test harnesses.

> **Note**
>
> This section applies to middleware implementations of Oracle Utilities Smart Grid Gateway **only**. This section does NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.


- Test Harness Design
- Locating the WSDL for the Test Harness
- Web Services


## Test Harness Design

Each test harness is divided into two main layers: a front end and back end layer. The front end set of services implements the head-end system interfaces, which receive requests corresponding to the specifications shown in the list below. Each of these services calls into the "back end" layer of the test harness, which defines meters and sets their attributes. These meters are stored in a file within the test harness called meterdb.xml.

> **Note**
>
> The meterdb.xml file can be modified pre-deployment, but not post-deployment. However, the test harness retains an in-memory "database" of the meters in the file. The in-memory representation can be modified using the utility web services. Note that any changes to the in- memory structure will be lost when the server is restarted or the test harness composite is redeployed.

**Itron OpenWay**

- www.itron.com.ami.2008.10.control
  - ReconnectMeter
  - GetReconnectMeterResult
  - DisconnectMeter
  - GetDisconnectMeterResult
- www.itron.com.ami.2008.10.data
  - ContingencyReadByEndpoint
  - GetContingencyReadByEndpointResult
  - InterrogateByGroup
  - GetInterrogateByGroupResult



Copyright © 2026, Oracle and/or its affiliates.

- www.itron.com.ami.2008.10.control
  - PingByEndpoints
  - ReconnectMeter
  - GetReconnectMeterResult
  - DisconnectMeter
  - GetDisconnectMeterResult
- www.itron.com.ami.2012.03.control.diagnostic
  - PingByEndpoints
  - GetPingByEndpointsResult
- www.itron.com.ami.2009.08.provisioning
  - AddMeterDefinitions
  - DeregisterMeters

**Landis+Gyr**

- http://www.multispeak.org/Version_3.0 MR_CB (Meter Reading_Customer Billing)
  - MeterAddNotification
  - MeterRemoveNotification
  - InitiateMeterReadByMeterNumber
- http://www.multispeak.org/Version_3.0 CD_CB (Connect/Disconnect_Customer Billing)
- InitiateConnectDisconnect

The LG Harness will send below responses to corresponding BPEL composites:

- http://www.multispeak.org/Version_3.0 CB_MR (Customer Billing_Meter Reading)
- ReadingChangedNotification
- http://www.multispeak.org/Version_3.0 CB_CD (Customer Billing_Connect/Disconnect)
- CDStateChangedNotification

**Sensus RNI**

- http://www.multispeak.org/Version_4.1_Release MR_Server
  - MeterAddNotification
  - MeterRemoveNotification
  - InitiateMeterReadingsByMeterID
- http://www.multispeak.org/Version_4.1_Release CD_Server
  - IitiateConnectDisconnect
- http://www.multispeak.org/Version_4.1_Release CB_Server
  - CDStatesChangedNotification (async reply)
  - ReadingChangedNotification (async reply)
- http://www.multispeak.org/Version_4.1_Release OD_Server
  - InitiateOutageDetectionEventRequest

**Silver Spring Networks**


Copyright © 2026, Oracle and/or its affiliates.

- urn:com:ssn:schema:service:v1.4:DataAggregation and urn:com:ssn:schema:service:v1.6:DataAggregation
  - getMeterFieldStatus
- urn:com:ssn:schema:service:v1.4:DeviceManager and urn:com:ssn:schema:service:v1.6:DeviceManager
  - FindDevice
  - ReplaceDeviceAtLocation
  - ReplaceLocation
- urn:com:ssn:schema:service:v1.4:DeviceResults and urn:com:ssn:schema:service:v1.6:DeviceResults
  - getRemoteProvisioningResultsByJobID
  - getMeterReadResultsByJobID
  - getPingResultsByJobID
- urn:com:ssn:schema:service:v1.4:JobManager and urn:com:ssn:schema:service:v1.6:JobManager
  - addRemoteProvisioningJob
  - getJobStatus
  - addMeterReadJob
  - addPingJobfindJob
  - getJobStatusForDevice

## Locating the WSDL for the Test Harness

Follow these procedures to locate the test harness web service definition language (WSDL):

**How to Use Enterprise Manager to Locate the WSDL**

1. Open Enterprise Manager and use the navigation pane to open the dashboard of the test harness composite:
2. The top bar of the dashboard contains several buttons and icons. One of these is a “world” icon with a puzzle piece over it. Click this icon to display a list of the WSDLs and endpoint URIs for the composite:
3. Click the UtilService WSDL URL link to see the WSDL in the browser, or right click and save it to your machine

Depending on your requirements, it may be necessary to download the associated schema found in the wsdl:types section. The URL can be pasted into a browser tab and downloaded in the same manner as the WSDL. The main schema has imported schemas that may also be required.

**How to Use a Direct URL to Locate the WSDL**

The WSDL can be accessed without Enterprise Manager by understanding the paths used on the SOA server. In general, they have the following form:

```
http://{server name}:{port number}/soa-infra/services/{partition}/{Composite}/
{Web Service}?WSDL
```


Copyright © 2026, Oracle and/or its affiliates.

For example, the test harness WSDL can be found at the following locations for each SGG adapter.

**Itron OpenWay**:

```
http://{server name}:{port number}/soa-infra/services/Itron_Test/
ItronTestHarness/UtilService? WSDL
```

**Landis+Gyr**:

```
http://{server name}:{port number}/soa-infra/services/LG_Test/LGTestHarness/
UtilService?WSDL
```

**Sensus RNI**:

```
http://{server name}:{port number}/soa-infra/services/Sensus_Test/Sensus/
UtilService?WSDL
```

**Silver Spring Networks**:

```
http://{server name}:{port number}/soa-infra/services/SSN_Test/SSNTestHarness/
UtilService?WSDL
```

## Web Services

This section describes the web services included in the test harness BPEL composite for each supported head-end system.

- General Services
- Locate Meter Services
- Meter Administration Services
- Meter Attribute Administration Services

### General Services

This section describes the general services of the test harness composite.

#### LoadMeterIndex

This web service loads the data store from the internal file. By default, if the store is already in memory, it will NOT reload. This behavior can be overridden with the forceReload parameter.

Input — LoadMeterIndexInput

**Part:** payload

**Element:** LoadMeterIndexRequest

| Parameter | Description |
|---|---|
| forceReload | A switch telling the system whether to reload the meter index from the configuration file. Default is false. |


Copyright © 2026, Oracle and/or its affiliates.

Output — LoadMeterIndexOutput

**Part:** payload

**Element:** LoadMeterIndexResult

| Parameter | Description |
|---|---|
| loaded | A boolean value for whether or not the index was reloaded from the configuration file |

Fault — UtilityFault (see UtilityFault for more details).

#### ViewAuditTrail

This web service returns the audit log for the entire session.

Input — ViewAuditTrailInput

**Part:** payload

**Element:** ViewAuditTrailRequest

Output — ViewAuditTrailOutput

**Part:** payload

**Element:** ViewAuditTrailResult

An Entry consisting of a timestamp and an Operation. Each entry may have an associated meter object showing the latest update.

Fault — See UtilityFault, above.

#### UtilityFault

Fault with similar mapping to SGG/OUAF faults:

Typically, the faultCode, faultString, faultActor, and detail/text elements will be populated.

## Locate Meter Services

This section describes the locate meter web services of the test harness composite.

#### FindMeters

This web service queries the data store for one or more meters. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input — FindMetersInput

**Part:** payload

**Element:** FindMetersRequest

| Parameter | Description |
|---|---|
| id | The meter ID for which to search |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Description |
|---|---|
| isRegex | The provided id can be a regex value when this parameter is true. Hint: to search for all meters in the system, use ".*" for the ID. |

Output — FindMetersOutput

**Part:** payload

**Element:** FindMetersResult

Zero or more meter objects can be returned from the search

Fault — See UtilityFault. Unlike other methods, FindMeters does not throw an exception if the meter is not found. As such, it can be used to test for the existence of a Meter prior to querying for it.

### IsMeterDefined

This web service queries whether a particular meter is defined in the data store.

Input — IsMeterDefinedInput

**Part:** payload

**Element:** IsMeterDefinedRequest

| Parameter | Description |
|---|---|
| id | The meter ID for which to search |

Output — IsMeterDefinedOutput

**Part:** payload

**Element:** IsMeterDefinedResult

Whether or not the provided ID is part of the index.

Fault — See UtilityFault. Thrown when meter id is not found.

### GetMeter

This web service returns all the attributes of a single meter from the in-memory data store. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input — GetMeterInput

**Part:** payload

**Element:** GetMeterRequest

| Parameter | Description |
|---|---|
| id | The meter ID for which to search |

Output — GetMeterOutput


Copyright © 2026, Oracle and/or its affiliates.

**Part:** payload

**Element:** GetMeterResult

The meter object requested by the ID.

Fault — See UtilityFault. Thrown when meter id is not found.

## Meter Administration Services

This section describes the meter administration services of the test harness composite.

### AddMeters

This web service adds a set of meters to the in-memory data store. This will not permanently add it to the control file.

Input — AddMetersInput

**Part:** payload

**Element:** AddMetersRequest

| Parameter | Description |
|---|---|
| id | The identification code for the meter. |
| macID | A MAC address that must be unique within the system. |
| utility | An informational string. |
| serviceType | One of the valid ServiceType values (see schema). "Electric" is the only option at this time. |
| isCommissioned | Whether or not the meter is in a commissioned state. |
| loadActionCode | One of the possible LoadActionCode values used in Connect and Disconnect (see schema). |
| outageEventType | One of the possible OutageEventType values used in Device Status Check (see schema). |
| executionStatus | One of the possible ExecutionStates (see schema). These values control how the meter will respond to commands. |
| groupName | The name linking multiple meters together into a set. |
| jobExecutionStatus | One of the possible Job Execution Status values (see schema). This attribute determines how requested jobs perform. |
| updateIfExisting | Whether or not to update the meter with the provided values if it already exists in the index. |
| Comment | An informational string describing the purpose of the meter. |
| Channels | A listing of unit of measures supported by this meter. |
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |
| decimals | The number of digits to the right of the decimal that should be generated when reading the meter. |
| description | A longer description of the unit of measure. |

Output — AddMetersOutput

**Part:** payload


Copyright © 2026, Oracle and/or its affiliates.

**Element:** AddMetersResult

Whether or not each meter was added to the index.

Fault — See UtilityFault

### RemoveMeter

This web service removes a meter from the in-memory data store. This will not permanently remove it from the control file.

Input — RemoveMeterInput

**Part:** payload

**Element:** RemoveMeterRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter to be removed |

Output — RemoveMeterOutput

**Part:** payload

**Element:** RemoveMeterResult

Whether or not the meter was removed from the index.

Fault — See UtilityFault. Thrown when meter id is not found.

### AddMeterChannel

This web service adds a new channel to a single meter.

Input — AddMeterChannelInput

**Part:** payload

**Element:** AddMeterChannelRequest

| Parameter | Description |
|---|---|
| id | The identification code for the meter. |
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |
| decimals | The number of digits to the right of the decimal that should be generated when reading the meter. |
| description | A longer description of the unit of measure. |

Output — AddMeterChannelOutput

**Part:** payload

**Element:** AddMeterChannelResult

Whether or not the channel was added to the index.

Fault — See UtilityFault. Thrown when meter id is not found.


Copyright © 2026, Oracle and/or its affiliates.

### RemoveMeterChannel

This web service removes a Channel from a meter.

Input — RemoveMeterChannelInput

**Part:** payload

**Element:** RemoveMeterChannelRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter to be removed. |
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |

These three parameters are combined to locate a unique channel

Output — RemoveMeterChannelOutput

**Part:** payload

**Element:** RemoveMeterChannelResult

Whether or not the channel was removed from the meter.

Fault — See UtilityFault. Thrown when meter id is not found.

### ReadScalarMeter

This web service generates a scalar reading for each channel of a given meter.

Input — ReadScalarMeterInput

**Part:** payload

**Element:** ReadScalarMeterRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter to be read |

Output — ReadScalarMeterOutput

**Part:** payload

**Element:** ReadScalarMeterResult

Zero or more scalar readings for the given meter.

| Parameter | Description |
|---|---|
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |
| decimals | The number of digits to the right of the decimal that should be generated when reading the meter. |
| description | A longer description of the unit of measure. |
| value | A random number representing the scalar reading. |


Copyright © 2026, Oracle and/or its affiliates.

## Meter Attribute Administration Services

This section describes the meter administration services of the test harness composite.

### GetLoadActionCode

This web service queries whether the given meter is connected or disconnected. This method is used by the Connect/Disconnect service. The values for load action code are:

- Connect
- Disconnect

Input — GetLoadActionCodeInput

**Part:** payload

**Element:** GetLoadActionCodeRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the load action code status should be retrieved |

Output — GetLoadActionCodeOutput

**Part:** payload

**Element:** GetLoadActionCodeResult

The connection status of the requested meter.

Fault — See UtilityFault. Thrown when meter id is not found.

### SetLoadActionCode

This web service updates the load action code for a given meter. This method is used by the Connect/Disconnect service. The values for load action code are:

- Connect
- Disconnect

Input — SetLoadActionCodeInput

**Part:** payload

**Element:** SetLoadActionCodeRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the load action code status should be set. |
| value | The new value of LoadActionCode to set on the meter. |

Output — SetLoadActionCodeOutput

**Part:** payload

**Element:** SetLoadActionCodeResult


Copyright © 2026, Oracle and/or its affiliates.

The boolean response indicates the success or failure of the update (not the current field status).

Fault — See UtilityFault. Thrown when meter id is not found.

### IsCommissioned

This web service queries the commissioning status for a given meter. This service is used by the Commission/Decommission process. The commissioning attribute can be true or false.

Input — IsCommissionedInput

**Part:** payload

**Element:** IsCommissionedRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the Commissioned status should be retrieved |

Output — IsCommissionedOutput

**Part:** payload

**Element:** IsCommissionedResult

The value of the Commissioned status attribute for the requested meter.

Fault — See UtilityFault. Thrown when meter id is not found.

### SetCommission

This web service updates the commissioning status for a given meter. This service is used by the Commission/Decommission process. The commissioning attribute can be true or false.

Input — SetCommissionedInput

**Part:** payload

**Element:** SetCommissionedRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the Commissioned status should be set |
| value | The new value of Commissioned status to set on the meter |

Output — SetCommissionedOutput

**Part:** payload

**Element:** SetCommissionedResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault — See UtilityFault. Thrown when meter id is not found.


Copyright © 2026, Oracle and/or its affiliates.

### GetExecutionStatus

This web service queries the status of the property controlling the overall execution of the command. The possible values of execution status are:

- Success - The command should complete successfully
- ResponseTimeout - The asynchronous response will never arrive
- SyncOperationFail - A simulated fault will occur in the during the initial request
- AsyncOperationFailure - A simulated fault will occur in the asynchronous response

Input — GetExecutionStatusInput

**Part:** payload

**Element:** GetExecutionStatusRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the ExecutionStatus should be retrieved |

Output — GetExecutionStatusOutput

**Part:** payload

**Element:** GetExecutionStatusResult

The value of the ExecutionStatus attribute for the requested meter.

Fault — See UtilityFault. Thrown when meter id is not found.

### SetExecutionStatus

This web service updates the property controlling the overall completion of the command. The possible values of execution status are:

- Success - The command should complete successfully
- ResponseTimeout - The asynchronous response will never arrive
- SyncOperationFail - A simulated fault will occur in the during the initial request
- AsyncOperationFailure - A simulated fault will occur in the asynchronous response

Input — SetExecutionStatusInput

**Part:** payload

**Element:** SetExecutionStatusRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the ExecutionStatus should be set |
| value | The new value of ExecutionStatus to set on the meter |

Output — SetExecutionStatusOutput

**Part:** payload

**Element:** SetExecutionStatusResult


Copyright © 2026, Oracle and/or its affiliates.

The boolean response indicates the success or failure of the update (not the current field status).

Fault — See UtilityFault. Thrown when meter id is not found.

# Creating a Custom Adapter for Smart Grid Gateway

Customers can create their own customized adapter for using Smart Grid Gateway with a specific head-end system. This can be done using the Oracle Utilities Smart Grid Gateway Adapter Development Kit. This section describes the Adapter Development Kit and its components.

> **Note**
>
> Unless otherwise noted, the contents of this section apply to middelware implementations of Oracle Utilities Smart Grid Gateway **only**, and do NOT apply to native implementations or Oracle Utilities cloud services. See Smart Grid Gateway Adapter Native Implementations for more information about Smart Grid Gateway Adapter native implementations and implementing Smart Grid Gateway adapters with Oracle Utilities cloud services.

- Adapter Development Kit Overview
- Adapter Development Kit Processing
- Oracle Service Bus Processing
- Business Processing Execution Language Processing
- Configuring an Adapter Development Kit Head-End System
- Configuring Adapter Development Kit Extendable Lookups
- Using the Adapter Development Kit Test Harness
- The Adapter Development Kit Native Format
- Adapter Development Kit Native Format Example
- Adapter Development Kit Native Format Schema

## Adapter Development Kit Overview

The Oracle Utilities Smart Grid Gateway (SGG) Adapter Development Kit provides a starting point for customers to create their own customized adapter for using SGG with a specific head-end system. SGG uses Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL) to facilitate communication with other head-end systems. The following functionality can be configured:

**Measurement Data and Device Event Loading**: Data parsing and transformation via Oracle Service Bus from the smart meter format into the Oracle Utilities Service and Measurement Data Foundation unified format for measurement data and device events.

**Measurement Data and Device Event Processing**: Configurable mapping for meter status codes and device event names to Oracle Utilities Service and Measurement Data Foundation standard values.


Copyright © 2026, Oracle and/or its affiliates.

**Commissioning Communication**: Business objects and BPEL processes to support the Meter Add Notification message.

**Connect Communication:** Business objects and BPEL processes to support Initiate Connect/Disconnect and Connect/Disconnect State Change Notification messages.

**Disconnect Communication**: Business objects and BPEL processes to support Initiate Connect/Disconnect and Connect/Disconnect State Change Notification messages.

**Decommissioning Communication**: Business objects and BPEL processes to support Meter Removal Notification message.

**On-Demand Read**: Business objects and BPEL processes to support Initiate Meter Read by Meter Number and Reading Changed Notification messages.

**Device Status Check**: Business objects and Oracle BPEL processes to support the outbound Initiate Outage Detection communication, the inbound Outage Detection Event Notification communication, and the processing of Event Notifications.

#### What Does the Adapter Development Kit Provide?

The Oracle Utilities Smart Grid Gateway Adapter Development Kit includes the following:

**Sample Oracle Service Bus Processes**: A file parsing and transformation process framework, delivered as an Oracle Service Bus (OSB) configuration, that can be used as a reference for creating the functionality to import usage readings and device events from a specific head-end system.

**Sample Oracle Business Process Execution Language Processes**: A set of sample communication processes that support Remote Connect, Remote Disconnect, Commission, Decommission, Device Status Check, and On-Demand Read commands delivered as an Oracle Business Process Execution Language (BPEL) project.

**Oracle Utilities Smart Grid Gateway Business Objects:** Business objects to support measurement data and device event loading and two-way commands such Commission, Decommission, Remote Connect, Remote Disconnect, and Device Status Check.

**Sample Two-Way Communication Test Harness**: A head-end system emulator delivered as a BPEL composite.

**Demonstration Data**: A set of sample data provided in the demonstration database that can be used with the adapter processes as delivered (prior to any customization you might perform as part of your implementation).

## Adapter Development Kit Processing

This section provides details concerning the OSB processing, BPEL Processes, and OUAF objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any customizations you create as part of your implementation.


- Initial Measurement Data and Device Event Loading
- Device Communication


### Initial Measurement Data and Device Event Loading

The initial measurement data load and subsequent device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway



Copyright © 2026, Oracle and/or its affiliates.

service format. Payloads contain measurements and meter events in some head-end specific format OSB then places each service call into a JMS queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel then a service creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format.

- Initial Measurements and Device Events
- Base Package Business Objects

## Initial Measurements and Device Events

The usage and event data exported from the AMI head-end system is loaded into Oracle Utilities as initial measurement and device event data. You can customize processing of this by configuring the following base product OSB projects:

1. **SGG-DG-CSV-BASE** contains components that are not to be changed on customer site. They implement functions specific to the CSV format processing such as validation and transformation.
2. **SGG-DG-CSV-CM** allows for customization and simplifies future upgrades.
3. **SGG-DG-XML-BASE** contains components that are not to be changed on customer site. They implement functions specific to the IMD and event upload format processing such as validation and transformation.
4. **SGG-DG-XML-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-DG-CSV-CM project are stored in the EnvironmentSettings.xq xquery file. You can use this file to adjust initial measurement and device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the SGG-DG-CSV-CM EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| populateRaw | Determines if the initial measurement data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElementEvent | Holds the name of inbound web service for the device event seeder. | |
| destinationRootElementIMD | Holds the name of inbound web service for the IMD seeder. | |

The runtime configuration settings for the SGG-DG-XML-CM project are stored in the EnvironmentSettings.xq xquery file. You can use this file to adjust initial measurement and device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the SGG-DG-XML-CM EnvironmentSettings.xq file:


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| populateRawIMD | Determines if the initial measurement data is populated as raw data. | true<br>false |
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElement | Holds the name of inbound web service for the IMD seeder. | |

For additional information about the sample OSB implementation included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit, see Oracle Service Bus Processing.

### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the SGG-DG-CSV-CM and SGG-DG-XML-CM projects. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

### Processing Data in the Adapter Development Kit Native Format

Usage and event data exported from the AMI head-end system can be loaded into Oracle Utilities in the “native” initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects). You can customize processing of this by configuring the following base product OSB projects:

1. **SGG-DG-SEEDER-BASE** contains components that are not to be changed on customer site. They implement functions specific to the IMD and event upload format processing such as validation and transformation.
2. **SGG-DG-SEEDER-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-DG-SEEDER-CM project are stored in the EnvironmentSettings.xq xquery file. You can use this file to adjust initial measurement and device event data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the SGG-DG-SEEDER-CM EnvironmentSettings.xq file:

| Element | Description | Valid Values |
|---|---|---|
| callPreProcessing | Determines if the preprocessing proxy service is called. | true<br>false |
| callPostProcessing | Determines if the postprocessing proxy service is called. | true<br>false |
| destinationRootElement Interval | Holds the name of inbound web service for the interval IMD seeder. | |


Copyright © 2026, Oracle and/or its affiliates.

| Element | Description | Valid Values |
|---|---|---|
| destinationRootElement Scalar | Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval. | |
| destinationRootElement Event | Holds the name of inbound web service for the device event seeder. | |
| publishServices/service | Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker). | |
| filterEvents | Determines if events should be filtered. | true<br>false |
| filterUsage | Determines if usage should be filtered. | true<br>false |

See The Adapter Development Kit Native Format for more information about the ADK “native” format.

### Publishing Initial Measurement Data and Device Events

The Adapter Development Kit can be configured to publish initial measurement data and device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

### Enabling Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-DG-SEEDER-CM OSB project are used in publishing measurement data and device events to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

### Configuring Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems.

Initial measurement and device event data is published in the “native” initial measurement data format (the format of the initial measurement and device event seeder business objects). This format includes normalized unit of measure, condition, and device event codes. See the *Oracle*


Copyright © 2026, Oracle and/or its affiliates.

*Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published via this feature is published prior to VEE processing. In addition, filtering can NOT be applied to initial measurement or device event data published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which Oracle DataRaker will consume the converted XML files. | /spl/sploutput/int |

### Filtering Initial Measurement Data

The Adapter Development Kit can be configured to filter initial measurement data passed into Oracle Utilities Smart Grid Gateway and Meter Data Management. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```


Copyright © 2026, Oracle and/or its affiliates.

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **ADK - UOM Code to Standard UOM Mapping** extendable lookup (DG-HeadendUOMLookup) are passed into the system for processing.

### Filtering Events

The Adapter Development Kit can be configured to filter device events passed into Oracle Utilities Smart Grid Gateway and Meter Data Management. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **ADK - Device Event Mapping** extendable lookup (DG-DeviceEventMappingLookup) are passed into the system for processing.

### Prioritized Device Event Processing

The SGG adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (D1RT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing. See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

### Configuring Payload Processing Notifications

By default, Payload Statistics, Payload Error Notification, and Payload Summary notification activities are sent to the Notification Queue within Oracle Service Bus. Smart Grid Gateway adapters can be configured to restrict sending these activities by defining optional JCA transport dynamic properties in the appropriate OSB project. The following optional JCA transport dynamic properties can used to control if these notification activities are sent:

- **SGG_SEND_STATISTICS**: Controls sending Payload Statistic activities (true: send, false: do not send)
- **SGG_SEND_ERROR**: Controls sending Payload Error Notification activities (true: send, false: do not send)
- **SGG_SEND_SUMMARY**: Controls sending Payload Summary activities (true: send, false: do not send)

When these properties are set to false, the corresponding payload notification will not be sent to the JMS Queue. Note these properties are not populated by default, and default to true if omitted.

To create these properties:

1. Navigate to the SGG-D3-CIM-EVENT-CM project in the Oracle Service Bus Console.
2. Expand the **Proxy Services** folder.
3. Select the **InboundProxyService** proxy service (not the Pipeline or WSDL).
4. Click **Create** to create a new session.
5. Select **Transport Details**.
6. Click the plus sign in the **Dyanmic EndPoint Properties** section. An empty row will appear.
7. Enter the property you wish to define in the **Property** column.


Copyright © 2026, Oracle and/or its affiliates.

8. Enter "false" in the **value** colulmn.
9. Repeat steps 6-8 for each property you wish to define.
10. Click **Activate** to save and activate your changes.

> **Note**
>
> These properties are only used if the SGG_EXT_ID_PATTERN is not present or the regular expression captures nothing. The SGG_EXT_ID_PATTERN will override these properties when the regular expression captures group data.

## Base Package Business Objects

The adapter development kit base package includes the following device and initial measurement business objects:

| Business Object Name | Description |
|---|---|
| DG-InitialLoadIMDInterval | Generic Initial Load IMD - Interval<br>Used when loading customized adapter interval measurements into the system for the first time. |
| DG-InitialLoadIMDScalar | Generic Initial Load IMD - Scalar |
| DG-SmartMeter | ADK — Smart Meter |

# Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, device status check, or an on-demand read. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related head-end system web service. The head-end system then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

- Communication Flows
- Device Communication Base Package Business Objects
- External System
- Inbound / Outbound Service Configuration
- BPEL Processes

## Communication Flows

The table below lists the communications created for each Adapter Development Kit command:


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Remote Connect | ADK - Initiate Connect or Disconnect | ADK - Connect/ Disconnect State Change | Connect Device<br>Create IMD |
| Remote Disconnect | ADK - Initiate Connect Disconnect | ADK - Connect/ Disconnect State Change | Disconnect Device<br>Create IMD |
| Device Commissioning | ADK - Meter Add Notification | | Device Commissioning |
| Device Decommissioning | ADK - Meter Remove Notification | | Device Decommissioning |
| On-Demand Read (Scalar) | ADK - Initiate Meter Read by Meter ID | ADK - Reading Changed Notification | Create IMD |
| On-Demand Read (Interval) | ADK - Initiate Meter Read by Meter ID | ADK - Reading Changed Notification | Create IMD |
| Device Status Check | ADK - Initiate Outage Detection | ADK - Outage Detection Event Notif | |

### Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the DG-CCOMPEV (Create IMD Completion Event) algorithm on the Create Completion Events status of the DG-ReadingChgNotification (ADK - Reading Changed Notification) business object.

The DG-VALDVCFIL (Check if Device has any Filtering errors) algorithm on the Validate status of the DG-ReadingChgNotification (ADK - Reading Changed Notification) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the DG-CCOMPEV algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

## Device Communication Base Package Business Objects

The adapter development kit base package includes the following communication business objects:

| Business Object Name | Description |
|---|---|
| DG-ConnectDisconStateChgNtf | Generic - Connect/Disconnect State Change |
| DG-InitiateConnectDisconnect | Generic - Initiate Connect Disconnect |
| DG-InitiateMeterByMeterId | Generic - Initiate Meter Read By Meter ID |
| DG-InitiateOutageDetection | Generic - Initiate Outage Detection |
| DG-MeterAddNotification | Generic - Meter Add Notification |
| DG-MeterRemoveNotifcation | Generic - Meter Remove Notification |


Copyright © 2026, Oracle and/or its affiliates.

| Business Object Name | Description |
|---|---|
| DG-OutageDetectEvtNotification | Generic - Outage Detection Event Notification |
| DG-ReadingChgNotification | Generic - Reading Changed Notification |

### Event Data Mapping

The head-end system event file format can map as follows into the business object, D1-DeviceEventMappingLookup:

| Head-End System Flat File Field | Device Event Seeder BO Element | Comments |
|---|---|---|
| Transaction ID (from Header record) | External Source Identifier | This is the file name. |
| Device Identifier | External Device Identifier | |
| Event Name | External Event Name | |
| Event Creation Date/Time | Event Date/Time | |
| Device Type | External Device Type | This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form):<br>Meter<br>Collector<br>Router |
| Service Location ID | External Service Location ID | |
| Communication Module Serial Number | External Communication Module Identifier | |
| Event Category ID | External Event Category | |
| Event Severity | External Event Severity | Valid values include (although the element itself is free-form):<br>Alert<br>Information |
| Status Value | External Status Value | This represents additional information that relates to the event itself. |
| Status Date/Time | External Status Date/Time | The date & time at which the additional information referenced above had occurred. |

## External System

You must create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)


Copyright © 2026, Oracle and/or its affiliates.

- Message XSL, W3C Schema, and Response XSL (as applicable)

## Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The adapter for Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

### Inbound Web Services

Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway Adapter Development Kit includes the following inbound web services:

| Inbound Web Service | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Used for upload of device events.<br>The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event.<br>If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created. |
| D1-InitialLoadIMD | Used for initial measurement upload.<br>The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system. |
| D1-DeviceStatusCheck | Device Status Check<br>This service is invoked by the integration layer to instantiate a Device Status Check command. |
| D1-InitialLoadIMD | Used by OSB to instantiate an IMD<br>This inbound web service is used by OSB to instantiate an Initial Measurement Data for incoming interval usage in the Generic format. |
| D1-RemoteConnect | Remote Connect<br>This service is invoked by the integration layer to instantiate a Remote Connect command. |
| D1-RemoteDisconnect | Remote Disconnect<br>This service is invoked by the integration layer to instantiate a Remote Disconnect command. |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service | Description |
|---|---|
| DG-ConDisconStChgNotification | Initiate Connect Disconnect response.<br>Retrieves asynchronous response from Initiate Connect Disconnect command. |
| DG-OutageDetectionEventNotification | Initiate Outage Detection Response<br>Retrieve response from the Initiate Outage Detection Event Notification command. |
| DG-ReadingChangedNotification | Reading Changed Notification<br>Notification that a device reading has changed. |

### Message Senders

Message senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway Adapter Development Kit includes the following message senders:

| Message Sender | Description |
|---|---|
| DG-COMM | Generic Adapter Commission |
| DG-DCOMM | Generic Adapter Decommission |
| DG-INTOUDET | Generic Adapter — Initiate Outage Detection Request |
| DG-XAISender | Commission / Decommission Sender |

## BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to MultiSpeak 4.1 format, invoking process callouts and invoking the remote endpoint to trigger the device events.

**OnDemandRead Composite Process**: Invokes the remote endpoint to trigger the on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

**ConnectDisconnect Composite Process**: Invokes the remote endpoint to trigger the connect/disconnect event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

**CommissionDecommission Composite Process**: Invokes the remote endpoint to trigger the commission or decommission event. After the synchronous call completes, a one of the following second business callout services is invoked to determined if the related “received” or “completed” callout should be executed:

- isExecutingCommissionReceived-Callout
- isExecutingCommissionCompleted-Callout
- isExecutingDecommissionReceived-Callout
- isExecutingDecommissionCompleted-Callout


Copyright © 2026, Oracle and/or its affiliates.

**DeviceStatusCheck Composite**: Invokes the remote endpoint to trigger the initiate outage detection event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

**ProcessCallout Composite**: This business callout provides a point at which customers and implementers can incorporate custom business logic and transformations. This composite includes the WSDLs and processing logic for all of the MultiSpeak processes. The default implementation of each method is a direct return of the input.

For additional information about the BPEL processes included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit, see Business Processing Execution Language Processing.

### Web Services

These web services are all defined in the head-end system. The WSDLs were added to a Meta Data Storage (MDS) layer in OUAF and all references to the WSDL point to this MDS location. These web services have HTTP security by default. You may need to modify the security as a part of your implementation.

| Web Service | Related BPEL Process | Description |
|---|---|---|
| CB_ServerService | ConnectDisconnect | This web service defines the return interface, the means by which the status is returned to the calling system.<br>This web service is only be invoked by the head end system, not OUAF. Only the CDStateChangeNotification web method is implemented in the composite.<br>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Generic Adapter/ConnectDisconnect/CB_ServerService |
| CB_Server | OnDemandRead | This web service defines the return interface, the means by which the reading is returned to the calling system.<br>This web service is only be invoked by the head end system, not OUAF. Only the ReadingChangedNotification web method is implemented in the composite.<br>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Generic Adapter/OnDemandRead/CB_Server |


Copyright © 2026, Oracle and/or its affiliates.

| Web Service | Related BPEL Process | Description |
|---|---|---|
| OA_ServerService | DeviceStatusCheck | This web service defines the asynchronous return for InitateOutageDetectionEventRequest for solicited responses.<br>This web service is only be invoked by the head end system, not OUAF. Only the ODEventNotification, PingURL, and GetMethods web methods are implemented in the composite.<br>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Generic Adapter/DeviceStatusCheck/OA_ServerService |

For additional information about the web services included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit, see Business Processing Execution Language Processing.

# Oracle Service Bus Processing

This chapter describes how to use Oracle Service Bus (OSB) and the components of the Oracle Utilities Smart Grid Gateway Adapter Development Kit to develop an adapter for importing usage reading and device events.

- OSB Overview
- Oracle Utilities Smart Grid Gateway Adapters

## OSB Overview

The examples in this chapter provide an implementer with the information required to create a customized adapter using the Oracle Utilities Smart Grid Gateway Adapter Development Kit. Customized adapters are required in situations where a productized Smart Grid Gateway adapter is not available for a particular head-end system. The examples below explain how to receive and process inbound meter usage and events. The examples contain two parts: the incoming file parsing functionality implemented in Java, and an easy, configurable and customizable Oracle Service Bus (OSB) configuration, including an implementation of post-parsing functionality.

- OSB Prerequisites
- OSB Related Documentation
- OSB Processing Terms and Acronyms

## OSB Prerequisites

Development of an Oracle Utilities Smart Grid Gateway adapter requires a number of technical skills as well as a development environment.

### Required Technical Skills

The following technical skills are required for developing an adapter using the development kit:


Copyright © 2026, Oracle and/or its affiliates.

- Experience in Java programming
- Experience in development using XQuery and XPath
- Experience in XML programming
- Experience in OSB development and administration
- Understanding of Oracle Utilities Service and Measurement Data Foundation (SMDF) architecture

**Environment Requirements**

Development of the Oracle Utilities Smart Grid Gateway adapter requires an environment configured as follows:

- The Oracle WebLogic server must be installed.
- The OSB must be installed.
- An OSB domain must be created.
- Java Messaging Service (JMS) queues for Initial Measurement Data (IMD) and Notification messages must be created.
- The spl-d1-osb-2.0.1.jar containing the com.splwg.d1.sgg.osb.common Java package must be available. This is a single jar for all vendor-specific adapters.
- The archive sgg-osb-generic-adapter.zip must be available. This file is downloaded as a part of adapter development kit deployment. It contains the FileParser folder with sample source code and the com.splwg.dg.sgg.osb.configuration.jar file with sample OSB projects.

## OSB Related Documentation

For more information about using the adapter development kit and Oracle Service Bus, see the following Oracle documents:

- *Oracle Fusion Middleware User's Guide for Technology Adapters*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*
- *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*
- *Oracle Fusion Middleware User's Guide for Oracle JDeveloper*
- *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide*
- *Oracle Utilities Meter Data Management / Smart Grid Gateway Administrative User Guide*

## OSB Processing Terms and Acronyms

This section lists several terms and acronyms used throughout this chapter.

| Term | Definition |
|---|---|
| AMI | Advanced Metering Infrastructure |
| CSV | Comma-Separated Values |
| FA | File Adapter. Oracle JCA File Adapter. |
| FP | File Parser, Java code that parses input file and generates Plain XML |
| GA | Smart Grid Gateway generic adapter |


Copyright © 2026, Oracle and/or its affiliates.

| Term | Definition |
|---|---|
| GFP | Generic File Processor, Java code that is invoked by FA. It communicates with FP |
| IMD | Initial Measurement Data |
| IMD XML | The final XML containing the IMD. |
| JAR | Java Archive |
| JAXB | Java Architecture for XML Binding |
| JCA | Java Connector Architecture |
| JDeveloper | Freeware Oracle integrated development environment for development of Java-based SOA and Java EE applications |
| JMS | Java Message Service |
| MDB | Message Driven Bean |
| SMDF | Service and Measurement Data Foundation |
| OSB | Oracle Service Bus |
| OUAF | Oracle Utilities Application Framework |
| Plain XML | An intermediate XML containing all of the values from the inbound file. It is necessary because OSB Message Flow can handle only XML. |
| PPS | Processing Proxy Service, OSB proxy service where the validation and transformation of Plain XML is implemented. It is necessary to catch errors that occurred before Plain XML is transformed to IMD or Device Event seeder structures. |
| RFD | Rejected File Descriptor, File containing information about a portion of input file that was a cause of error. |
| RPPS | Result Processing Proxy Service, OSB proxy service where processing of transformation result is implemented. It is necessary to catch errors that occurred after Plain XML is transformed to IMD or Device Event seeder structures. |
| SGG | Smart Grid Gateway |
| Weblogic | Oracle J2EE Application Server |
| WSD | Web Session Directory |
| XML | eXtensible Markup Language |
| XPath | Programming language for selecting nodes from an XML document |
| XQuery | Programming language to query XML. |

## Oracle Utilities Smart Grid Gateway Adapters

This section contains detailed information on the different components that make up an Oracle Utilities Smart Grid Gateway adapter and the application logic that is needed to implement one.

- Adapter Components
- Processing Life Cycle
- Java Project
- OSB Configuration
- Setting Up the Adapter Environment


Copyright © 2026, Oracle and/or its affiliates.

- Additional Information

## Adapter Components

The following table lists the adapter components:

| Component | Description |
|---|---|
| JCA File Adapter (FA) | Technology adapter for reading and writing files on the local file system. It is responsible for polling files from incoming folder and passing it to the Generic File Processor. |
| Generic File Processor (GFP) | Framework component that is implemented in Java. It is responsible for instantiation of the File Parser, getting Plain XML from it, and passing it to OSB message Flow. It also performs payload statistics related functionality such as gathering data and generating notification messages. |
| File Parser (FP) | Component implemented in Java. It is specific to every different incoming format. It is responsible for parsing the incoming file, breaking the payload into logical parts (debatching), generating Plain XML for every logical part, and returning XML back to invoking GFP.<br>**Plain XML** could be any logical subset of data that can be mapped to IMD. It is an intermediate format/schema between the raw data and IMD XML. The Plain XML schema must be defined by the file parser developer.<br>Raw data is read in portions from the input file and converted to Plain XML before passing to OSB message flow. The reason for data being read in portions is as follows:<br>Assume that an input file contains readings for 100 Measuring Components. You would want to read the input file in portions for two particular reasons.<br>1. Reading data of exactly one measuring component ensures that it is mapped appropriately to one IMD at a time.<br>2. Reading data in portions ensures that the entire file is not loaded into memory, which could cause resource issues. |
| Inbound Proxy Service | OSB proxy service that contains FA-related configuration settings. It statically routes all messages to Processing proxy Service. |
| Processing Proxy Service (PPS) | OSB proxy service that validates and transforms Plain XML. This service is necessary for catching errors that occur before Plain XML is transformed to IMD or Device Event seeder structures. |
| Result Processing Proxy Service (RPPS) | OSB proxy service that processes transformed data. This service is necessary for catching errors that occur after Plain XML is transformed to IMD or Device Event seeder structures. |

## Processing Life Cycle

This section outlines the life cycle from the initial input file to the initial measurement output. Refer to OSB Processing Terms and Acronyms for descriptions of the abbreviations and acronyms used in this chapter.


Copyright © 2026, Oracle and/or its affiliates.

See the **Logic Sequence Diagram** for general description of how the components interact.

**OSB Processes**

1. The JCA File Adapter starts reading the file.
2. FA instantiates and initializes the GFP.
3. FA invokes GFP passing an open stream to the incoming file.
4. GFP sends **D1-PayloadStatistics** notification message to FA.
5. GFP instantiates and initializes (if it is not done yet) the FP that is defined in the Inbound Proxy properties for parsing the file.
6. GFP invokes the FP (See File Parser Processes below).
7. In case of error in FP it invokes GFP. GFP sends **D1-PayloadErrorNotif** notification message to FA.
8. GFP gets the Plain XML as a return from FP.
9. GFP returns the Plain XML to the FA.
10. The Plain XML is passed via InboundProxyService to Processing Proxy service (PPS). In PPS's message flow, the Plain XML is validated and transformed to IMD XML.
11. PPS passes the IMD to Result Processing Proxy service (RPPS). RPPS publishes the IMD XML to a JMS queue, which is then picked up by Message Driven Bean (MDB).
12. OSB processes 3 through 9 are repeated until FP returns NULL on process 9.
13. GFP sends **D1-PayloadSummary** notification message to FA.

**File Parser Processes**

1. FP starts reading the input stream
2. FP instantiates a JAXB object of the type of Plain XML. The JAXB related interfaces and implementation classes are generated by using JDeveloper (See *Oracle Fusion Middleware User's Guide for Oracle JDeveloper* or Generating Java Classes from XML Schemas).
3. FP reads the input stream in segments until it hits a logical end of data with which it could create a Plain XML.
4. FP populates the data that is read into the Plain XML JAXB object.
5. FP marshals the JAXB object into Plain XML and passes back GFP. If the end of file (EOF) is reached FP passes back NULL.
6. File parser processes 2 through 5 are performed for every invocation by GFP.

## Java Project

The source code in the example files contains an implementation of functionality to parse an incoming payload in Comma Separated Values (CSV) and XML formats. This implementation can be used as a reference for creating a project that is specific to your head-end system. The descriptions below are mostly related to CSV parsing functionality as it is a more complex and interesting example. The XML related parsing example is provided only to illustrate how XML-based input formats may also be processed.

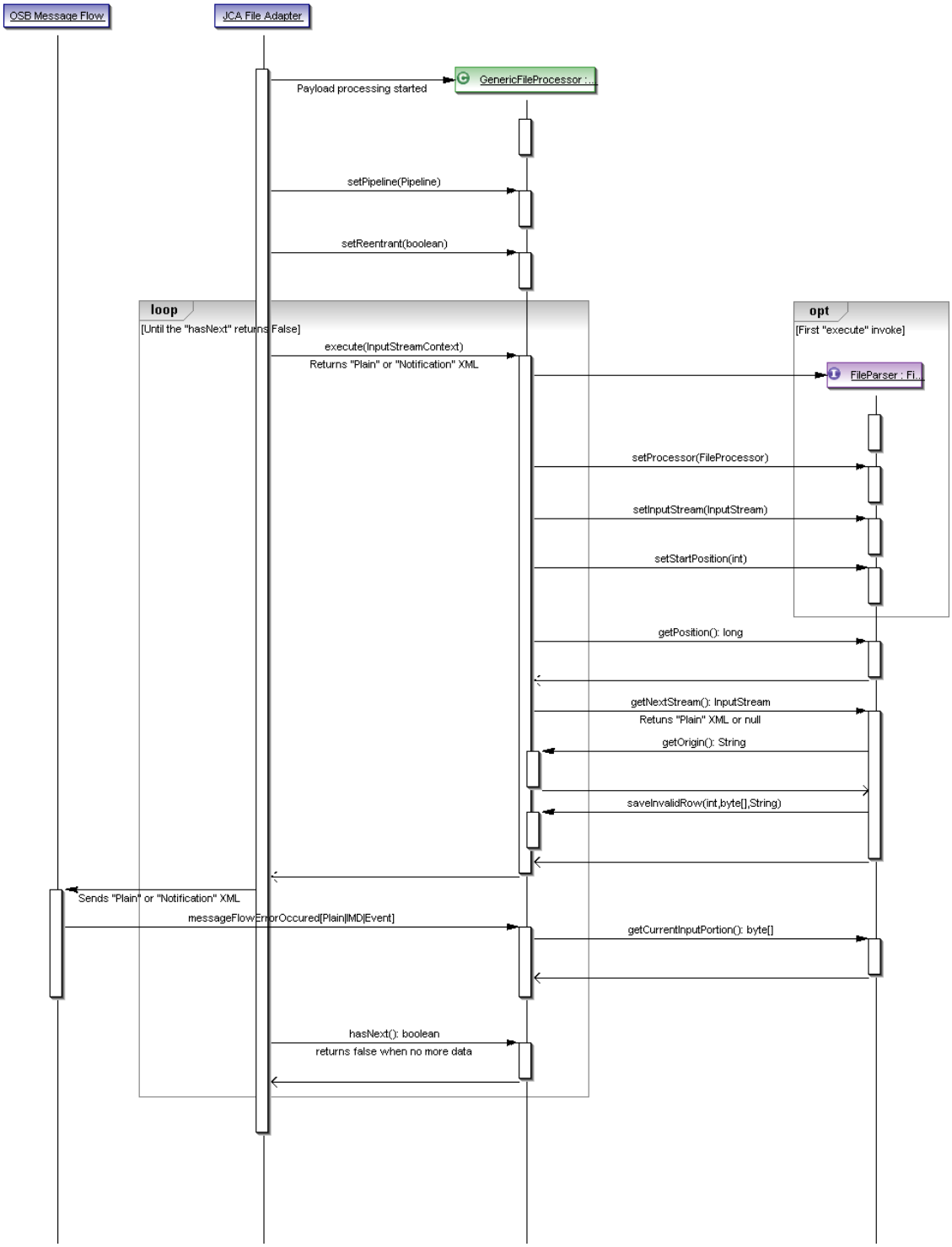
- Logic Sequence Diagram
- Java Implementation


Copyright © 2026, Oracle and/or its affiliates.

- Implementation Details

Logic Sequence Diagram

The sequence diagram below is helpful for understanding the File Parser functionality. The custom class must implement the FileParser and FileParser2 Java interfaces described in the SGG D1 jar to be qualify as a file parser. For more information, see FileParser Interface and FileParser2 Interface.





Copyright © 2026, Oracle and/or its affiliates.

When FA finds a file in the input directory, it instantiates and calls GFP which in turn starts an interaction with the FP. The interaction can be categorized into the following three phases:

1. Instantiation and Initialization
   - setProcessor()
   - setInputStream()
   - setStartPosition()
2. Transaction
   - getPosition()
   - getNextStream()
3. Exception Handling
   - getCurrentInputPortion()

### Instantiation and Initialization Phase

The GFP is instantiated and receives an Input Stream when a file gets into the input directory. At that point an instance of the FP is instantiated. Certain methods, as specified above, are invoked on the FP to initialize.

### Transaction Phase

Once the FP is initialized, GFP uses the FP instance to generate the Plain XML structure. This happens with the invocation of getNextStream() method on the FP.

On every invocation, the FP returns exactly one Plain XML. The FP invocations continues until NULL is returned from the FP. NULL indicates the end of file and an indication that parsing is complete for a specific file. GFP stops invoking getNextStream() for a particular input file when it receives a NULL from the FP.

FP uses the file input stream to read the file in portions. It could read byte-by-byte or line-by-line, but at the end of each getNextStream() method call it would have read only so much that is sufficient to create one Plain XML structure. For example, in an input file, if there is sufficient data for only one Plain XML then the first call to getNextStream() would return a Plain XML and the next one would return NULL. But if there is more than one, getNextStream() calls would continue to return Plain XML structures for every incoming portion until end of file is reached.

### Exception Handling Phase

Exception handling inside the FP can be categorized into two phases: **Reactive** and **Recovery:**

#### Reactive Phase

This phase involves catching an exception and reporting it to the GFP. It is achieved by invoking saveInvalidRow() method on the GFP (this.fileProcessor).

Note that when an exception occurs, no further parsing is done and NULL is passed back from getNextStream().

If one fails to return NULL, errors such as “Simultaneous good and bad result returned by the File Parser” can be found in the Weblogic server log.

When an exception is reported to the GFP, it does three things:


Copyright © 2026, Oracle and/or its affiliates.

1. Creates an XML payload D1-PayloadErrorNotif and passes it to the OSB message flow. The message is published to the Notification Queue from inside PPS of the BASE OSB project.
2. It increments the “transaction error occurred” variable by invoking the utility method inside the D1 jar. All the errors occurring during the lifetime of a file parsing are captured and are reported through the message D1-PayloadSummary - which is again published to Notification Queue.
3. Finally, it creates a data file in the error directory with the portion of raw data that is read when the error occurred. It also creates a Rejected File Descriptor (RFD) file. The raw input portion is available by utilizing getCurrentInputPortion() method, which is discussed below.

As noted in the above image, the saveInvalidRow() method takes three input parameters:

1. The position at which the parsing started. It will be written to an RFD file. This is helpful in identifying the location of an error in the input file.
2. The raw data that will be written into error data file. It should be in the same to the incoming file format and it should contain data that is read since current invocation of getNextStream() method is started. It will allow correcting troubled data and place “fixed” file into an incoming folder for further re-processing.
3. The error message that is reported in D1-PayloadErrorNotif message. Note, the content of the error message has to be defined by the file parser developer.

**Recovery Phase**

This phase must not be confused with recovering from a handled exception discussed above. This case is when file parsing gets completely interrupted by power failure or network interruption.

**Preparation for recovery:** GFP maintains an internal index to the current read position of the file. This is achieved by GFP by invoking getPosition() method on the FP before every getNextStream() method call.

```
public long getPosition() {
    long retVal = this.fileProcessor.getCurrentInputPosition();
    return retVal;
}
```

**Recovery:** In case of an interruption and subsequent restore, GFP sets the start position on the File Parser. For instance, if the failure had occurred at byte 415, the start position would be set to 415 so that FP starts to read from that point and not from 0. The value 415 can be referred to as the recovery point.

```
public void setStartPosition(long position) {
    this.startPosition = position;
}
```

When the file parser starts parsing again, two things must occur to ensure that it starts parsing from the recovery point:

1. Store the recovery point to the startPosition field.
2. Incorporate skip logic to start the pointer from the recovery point.

Java Implementation

The Java component of the project includes the following classes and packages:


Copyright © 2026, Oracle and/or its affiliates.

1. **com.splwg.dg.osb.common.FileParserGenGeneric** class is a super class with members that are common for CSV and XML related functionality. It implements the com.splwg.d1.sgg.osb.common.FileParser interface (see FileParser Interface). The implementation of com.splwg.d1.sgg.osb.common.FileParser2 (see FileParser2 Interface) has been added to define the IMD & Event Upload Statistics related functionality.
2. **com.splwg.dg.osb.csv.FileParserCSV** class contains the implementation of the parsing functionality for CSV formats. It performs the following logic:
    - Parses the incoming payload.
    - Transforms incoming data to Usage or Event related structures based on incoming data type.
    - Extends those structures with file information and external service provider identifier.
    - Returns one by one that structures as a stream to caller.
3. **com.splwg.dg.osb.plain** package contains JAXB related interfaces and implementation classes corresponding to the Plain XML Schema
4. **com.splwg.dg.osb.xml.FileParserXML** class contains the implementation of parsing functionality for the “IMD and Event Online Upload” XML format (See the *Oracle Utilities Service and Measurement Data Foundation User’s Guide*). It performs the following logic:
    - Parses the incoming payload.
    - Extracts information about current device.
    - Breaks the incoming data to separate initial measurement data structures.
    - Extends those structures with device information, file information and external service provider identifier.
    - Returns one by one the structures as a stream to caller.

Implementation Details

This section provides an overview of the implementation details. See the comments in the source code for more detailed information.

## OSB Configuration

This section outlines the OSB projects provided with the Adapter Development Kit.

- OSB Project Summary
- OSB Project Implementation Details

OSB Project Summary

The OSB Configuration consists of four projects. These projects can be categorized by either functionality or content:

**Functionality by Incoming File Format**

CSV format processing:

- SGG-DG-CSV-BASE
- SGG-DG-CSV-CM

XML format processing:

- SGG-DG-SEEDER-BASE


Copyright © 2026, Oracle and/or its affiliates.

- SGG-DG-SEEDER-CM
- SGG-DG-XML-BASE
- SGG-DG-XML-CM

### Content (by purpose of content)

Content containing business logic implementation:

- SGG-DG-CSV-BASE
- SGG-DG-SEEDER-BASE
- SGG-DG-XML-BASE

Content containing configuration settings related to the objects and variables required during the processing of payloads:

- SGG-DG-CSV-CM
- SGG-DG-SEEDER-CM
- SGG-DG-XML-CM

### Project Contents

The following table describes the contents of each of these projects:

| Project | Description |
|---|---|
| SGG-DG-CSV-BASE | Contains the components responsible for “actual” processing of data coming in CSV format. It can be upgraded in future releases without affecting the customization and environment settings that are stored in the SGG-DG-CSV-CM project. This project performs the following functions:<br>• Processes notification messages that are sent from the GFP.<br>• Calls the customizable local service proxies for pre- and post- processing of passed data<br>• Validates passed data against the XSD schema for Plain XML format<br>• Transforms passed Plain XML into D1-InitialLoadIMD or D1-DeviceEventSeeder formats according to type of incoming data.<br>• Sends result structures to destination JMS queue.<br>• Updates statistic information via calling the messageFlowIncResultIMD or messageFlowIncResultEvent method accordingly.<br>• Handles errors that occur to provide information to the File Processing component. Error handling contains separate handlers for two stages: before and after the Plain XML is transformed to the final XML structure. The separation is necessary to make Java calls to appropriate methods (messageFlowErrorOccurredPlain, messageFlowErrorOccurredIMD or messageFlowErrorOccurredEvent) according to the current processing stage and data type. |


Copyright © 2026, Oracle and/or its affiliates.

| Project | Description |
| --- | --- |
| SGG-DG-CSV-CM | Contains environment related configuration settings. It allows the customization and simplifies the future upgrades as well (See Configurng the OSB Project for example values). |
| SGG-DG-XML-BASE | Contains the components responsible for “actual” processing of data coming in XML format. It can be upgraded in future releases without affecting the customization and environment settings that are in SGG-DG-XML-CM project. This project performs the following functions:<br>• Processes notification messages that are sent from GFP.<br>• Calls the customizable local service proxies for pre- and post- processing of passed data.<br>• Validates passed data.<br>• Transform passed data. In current tutorial a simple XQuery code is used just to show the functionality available for transformation.<br>• Sends result structures to destination JMS queue in form of acceptable by the D1-DeviceEventSeeder and/or D1-InitialLoadIMD inbound web services.<br>• Updates statistic information via calling messageFlowIncResultIMD method.<br>• Handles errors to provides information to GFP. Error handling contains separate handlers for two stages: before and after the Plain XML is transformed to the final XML structure. The separation is necessary to make a Java calls to appropriate methods (messageFlowErrorOccurred or messageFlowErrorOccurredIMD) according to the current processing stage. |
| SGG-DG-XML-CM | Contains environment related configuration settings. It allows the customization and simplifies future upgrades as well. |


Copyright © 2026, Oracle and/or its affiliates.

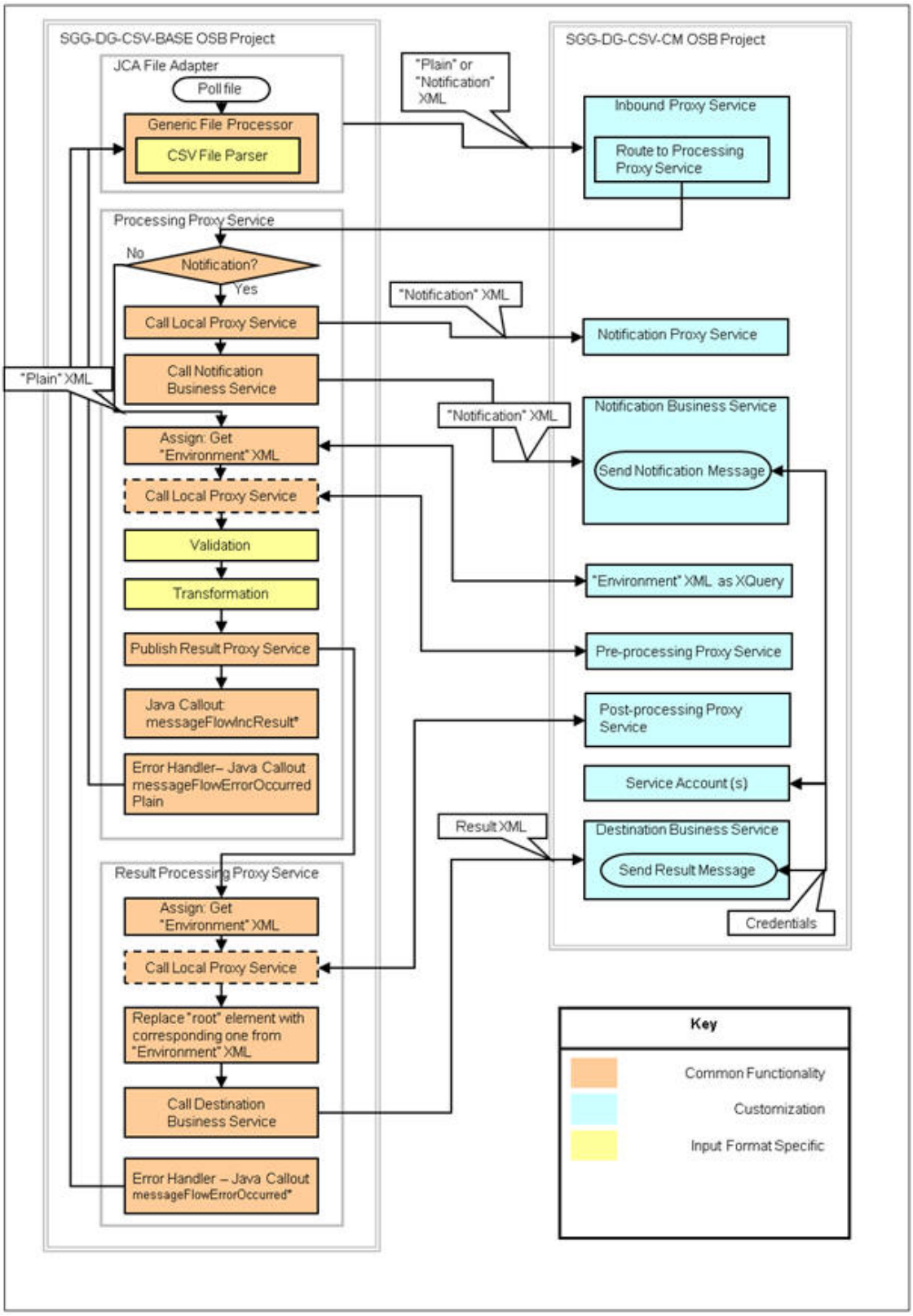
| Project | Description |
| --- | --- |
| SGG-DG-SEEDER-BASE | Contains the components responsible for processing of data coming in “native” XML format. It can be upgraded in future releases without affecting the customization and environment settings that are in SGG-DG-SEEDER-CM project. This project performs the following functions:<br>• Processes notification messages that are sent from GFP.<br>• Calls the customizable local service proxies for pre- and post- processing of passed data.<br>• Validates passed data.<br>• Splits payload into individual initial measurements and/or device events.<br>• Sends result structures to destination JMS queue in form of acceptable by the D1-InitialLoadIMD and/or D1-DeviceEventSeeder inbound web services.<br>• Updates statistic information via calling messageFlowIncResultIMD method.<br>• Handles errors to provides information to GFP. Error handling contains separate handlers for two stages: before and after the Plain XML is transformed to the final XML structure. The separation is necessary to make a Java calls to appropriate methods (messageFlowErrorOccurred or messageFlowErrorOccurredIMD) according to the current processing stage. |
| SGG-DG-SEEDER-CM | Contains environment related configuration settings. It allows the customization and simplifies future upgrades as well. |

## OSB Project Implementation Details

This section provides an overview of the implementation details. For more detailed information see the comments contained in the source code.

The following diagram shows the major processes and the data flow implemented in the OSB Configuration projects. The implementation description that follows is related to CSV parsing.


Copyright © 2026, Oracle and/or its affiliates.




## Setting Up the Adapter Environment

This section contains detailed information about the setting up of an environment with an example adapter.


Copyright © 2026, Oracle and/or its affiliates.

- Setting Up the Java Project
- Configurng the OSB Project


Setting Up the Java Project

1. Open the sgg-osb-generic-adapter.zip file and extract the FileParser folder into a temporary location on your hard disk.
2. Create a Java project in JDeveloper and import source code into it from the folder extracted previously.
3. Add the spl-d1-osb-2.0.1.jar file to the project's Libraries and Classpath properties.
4. Deploy the project as a jar file to the destination OSB domain's lib folder.

   **Note:** The WebLogic server should be restarted after jar file is copied to lib folder. For more details, see *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*.

Configurng the OSB Project

1. Open the sgg-osb-generic-adapter.zip file and extract the com.splwg.dg.sgg.osb.configuration.jar file into a temporary location on your hard disk.
2. Ensure that OSB is running in the domain that is created for the adapter.
3. Import resources by using the OSB Administration Console via the **System Administration > Import/Export > Import Resources** menu item from the extracted com.splwg.dg.sgg.osb.configuration.jar file. Note that conflicts may occur during import. They will be fixed in the following steps.
4. Check and change the JCA Transport Configuration properties of the SGG-DG-CSV-CM/Proxy Services/InboundProxyService proxy service according to the deployment environment:

   Where:

   - **Endpoint Properties/SGG_INPUT_PARSER** is a name of class that implements file parsing functionality. See a Java project mentioned early.
   - **Endpoint Properties/SGG_ERROR_FOLDER** is a folder where a rejected transaction from incoming file will be placed in case when the parsing or validation is failed.
   - **Endpoint Properties/SGG_SP_EXT_REF_ID** is a value corresponding to the target Service Provider's external reference id on OUAF side. It is used in the XML structures related to the IMD & Event Upload Stats functionality. Also, this value will be placed into D1-InitialLoadIMD/serviceProviderExternalId and D1-DeviceEventSeeder/externalSenderId elements.
   - **Activation Spec Properties/IncludeFiles** is the naming convention that the Oracle File Adapter uses to poll for inbound files.
   - **Activation Spec Properties/PhisicalArchiveDirectory** is a folder where to archive successfully processed files.
   - **Activation Spec Properties/PhisicalDirectory** is an input folder to be polled.
5. Change the Endpoint URI in the Transport Configuration section of the SGG-DG-CSV-CM/Business Services/DestinationBusinessService business service according to the deployment environment with value containing URI for JMS server and queue where IMD or Device Event structures will be sent (see *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus* for more detail.):



Copyright © 2026, Oracle and/or its affiliates.

6. Change the Endpoint URI in the Transport Configuration section of the SGG-DG-CSV-CM/Business Services/NotificationBusinessService business service according to the deployment environment:
7. Update the credentials stored in the SGG-DG-CSV-CM/Service Accounts/DestinationServiceAccount service account according to the deployment environment with user ID and password required to access JMS server:
8. Repeat steps 4-7 for the SGG-DG-XML-CM project

## Additional Information

This section contains additional information about the Adapter Development Kit.

- Processing Large Input Files
- FileParser Interface
- FileParser2 Interface
- FileProcessor Interface
- Generating Java Classes from XML Schemas

### Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

### FileParser Interface

This section provides the source code of the FileParser interface used by the Java class delivered with the Java package described earlier in this section.

```
package com.splwg.d1.sgg.osb.common;
import java.io.IOException;
import java.io.InputStream;
/** <p>
 *  FileParser component is responsible for processing the input
 *  stream and returning a Plain XML containing all data from
 *  incoming transaction in a shape expected by the following
 *  Oracle Service Bus message flow.
 * </p>
 */
public interface FileParser {
    /**
     * Returns the next message in Plain XML format
     * or null if there are no more messages
     * @return next message as InputStream
     */
    InputStream getNextStream() throws IOException;
    /**
     * Returns the current position in the currently
     * parsed source input stream
     * @return the position in source input stream after
     * last incoming row was read and parsed
```


Copyright © 2026, Oracle and/or its affiliates.

```
     */
    long getPosition();
    /**
     * Sets source stream
     * @param input the Input Stream that should be parsed
     */
    void setInputStream(InputStream input);
    /**
     * Sets "owner" FileProcessor
     * @param owner the Instance of FileProcessor that will
     * be notified about current processing via call to
     * utility methods (saveInvalidRow and setParseFinishNote)
     */
    void setProcessor(FileProcessor owner);
    /**
     * Sets the position from which the parsing of source
     * stream should be started.
     * It is necessary if last time the processing of
     * current file has been interrupted
     * @param position the starting position
     */
    void setStartPosition(long position);
    /**
     * Returns the byte array that contains a portion of an incoming
     * file that most recently has been read and converted to Plain XML.
     * It should be in the same format as incoming file. The header and
     * tail records, if there are, should be included and updated according
     * to the content
     */
    byte[] getCurrentInputPortion();
    /**
     * Returns type of currently generated transaction - USAGE or EVENT
     */
    FileProcessor.TransactionType getCurrentTransactionType();
    /**
     * Returns the number of successfully generated transactions of given
type
     */
    int getTransactionCount(FileProcessor.TransactionType type);
    /**
     * Returns the number of expected transactions in input stream
     * Returns -1 if information not available
     */
    int getTotalExpected();
}
```

## FileParser2 Interface

This section provides the source code of the FileParser2 interface used by the Java class delivered with the Java package described earlier in this section. This interface has been added to include a way for the File Processor to retrieve the file creation date which can be stored inside the incoming file.

```
package com.splwg.d1.sgg.osb.common;
import java.util.Date;
```


Copyright © 2026, Oracle and/or its affiliates.

```
public interface FileParser2 extends FileParser {
    /**
     * FileProcessor usage method
     * Returns the File creation date and time stamp stored
     * in the file
     * Returns null if information not available
     */
    Date getCreationDateTime();
}
```

FileProcessor Interface

This section contains a portion of the FileProcessor interface related to the file parsing functionality.

```
package com.splwg.d1.sgg.osb.common;
public interface FileProcessor {
    public enum TransactionType {
        USAGE, EVENT
    }
    /**
     * FileParser usage method
     *
     * @return a string containing the name of currently processed inbound
file
     */
    String getOrigin();
    /**
     * FileParser usage method Creates a file in the Error Folder and saves
     * passed row parameter to it
     *
     * @param position
     *            the number representing the position in file where invalid
     *            raw data is started
     * @param row
     *            the byte array containing an invalid raw data in vendor
     *            specific format the same with format of incoming file
     * @param failCause
     *            the string showing the cause of fail It will be used while
     *            logging the fail message.
     */
    void saveInvalidRow(long position, byte[] row, String failCause);
    /**
     * FileParser usage method
     * Returns the current offset in input file
     */
    long getCurrentInputPosition();
}
```

Generating Java Classes from XML Schemas

In JDeveloper you can use JAXB (Java Architecture for XML Binding) to generate Java classes from XML schemas. JAXB is an easy way to incorporate XML data and processing functions in Java applications without having to know XML. You can generate a JAXB 1.0 or 2.0 content model, including the necessary annotations, from an XML schema.


Copyright © 2026, Oracle and/or its affiliates.

When the JAXB binding compiler is run against an XML schema, JAXB packages, classes, and interfaces are generated. You can then use the generated JAXB packages and the JAXB utility packages in a binding framework to unmarshal, marshal, and validate XML content.

To generate Java classes from XML schemas with JAXB:

1. From the main menu choose **File > New > Business Tier > TopLink/JPA** and select either JAXB 1.0 or 2.0 Content Model from XML Schema to open the compilation dialog.
2. Select the schema file and optionally the JAXB customization file to use and the package to which the generated classes will be added.

The JAXB package and generated classes are added to the Application Resources folder.

# Business Processing Execution Language Processing

This section describes the Business Process Execution Language (BPEL) processes included in the Oracle Utilities Smart Grid Gateway Adapter Development Kit that support two-way communication between the adapter and the head-end system. The ideas and general practices described in this section can be applied to any other adapter.


- WSDLs, Endpoints, and Messages
- Composite Components
- Configuring and Customizing Adapter BPEL Processes
- Commands
- Working with Enterprise Manager
- MultiSpeak Implementation


## WSDLs, Endpoints, and Messages

WSDLs are Web Service Definition Language files that describe a web service. New services must utilize the WSDLs to determine the structure of requests and responses. For remote services, they contain the locations to access the services. For both remote and local (hosted) services, the files will contain the definitions for each web service including names, arguments, exceptions, and the structure of the input and output messages.

In Oracle Serivce Oriented Architecture (SOA), WSDLs can be classified as a “service” or as a “reference.” Services are hosted on the SOA server; that is, the implementation of the web service is on the local application server. Referenced services are implemented on a different server. This distinction is also relevant to the composite level. Services are implemented in the current composite. References are located elsewhere, possibly on the same app server, but in a different composite.

An “endpoint” is simply the URL for a web service. Since web services communicate via HTTP, each will have a unique URL. Once a service’s endpoint is known, a message service can be targeted to it.

### How to locate WSDLs and Endpoints

There are two ways to locate WSDLs in an installed Adapter: through Enterprise Manager and by using a direct URL. Only “hosted” WSDLs can be located in this way. Referenced (remote) WSDLs must be either located in source code or obtained from the hosted location.

### How to Use Enterprise Manager to Locate WSDLs and Endpoints:



Copyright © 2026, Oracle and/or its affiliates.

1. Open Enterprise Manager and use the navigation pane to open the Dashboard of the desired composite.
2. The top bar of the Dashboard contains several buttons and icons. One of these is an “earth” icon with a puzzle piece over it. Click this icon to display the WSDL and endpoint URIs for the composite.
3. Click the URL link to see the WSDL in the browser, or right click and save it to your machine for use in developing new services.

   Depending on the requirements, It may be necessary to download the associated schema or WSDL file(s). Schemas are available within the WSDL’s “types” element. Find associated WSDL URLs within the import element. That URL can be pasted into a browser tab.

   The endpoint for the service is also visible in this window. It is this URL that should be added to message sender configuration. For example:

   The value HTTP Header/SOA Action is taken from the service’s WSDL. Locate the wsdl:definitions/wsdl:binding/wsdl:operation/soap:operation element for the web method being invoked. An attribute called soapAction will contain the value for this field. A shortcut to this field is using Enterprise Manager's Testing framework (locate the Test tab from the dashboard view for the composite). Once the WSDL is parsed, the SOAP Action will appear in a field on the request tab:

   The values **HTTP Login User** and **HTTP Login Password** should be set to a valid WebLogic user that has access to the module. The **HTTP Method** should always be set to POST and the HTTP URL 1 should be set to the value of the endpoint above.

**How to Use a Direct URL to Locate WSDLs and Endpoints**

The WSDLs can be accessed without Enterprise Manager by understanding the paths used on the SOA server. In general, they have the form:

```
http://{server name}:{port number}/soa-infra/services/{partition}/{Composite}/
{Web Service}
```

## Composite Components

This section outlines several important logical features of the BPEL composites.

- Composite Properties
- Proxies
- Process Callouts

## Composite Properties

Most composites contain properties within the main file, composite.xml. They are essentially global constants that can be preconfigured with default values and changed at deployment time. They can be accessed post-deployment in Enterprise Manager.

Typically, these properties represent timeouts and boolean properties for activating or deactivating functionality. Other uses might include setting default values of constants such as URLs. They are accessed with the ora:getPreference() BPEL function.

They can be changed during development, at deployment time, or after deployment using different techniques for each.


Copyright © 2026, Oracle and/or its affiliates.

### Development Changes

At development time, properties can be created at will within the “component” sections of the composite.xml file. The image below shows an example of the properties found in DeviceStatusCheck:

```
<component name="DeviceStatusCheck">
  <implementation.bpel src="DeviceStatusCheck.bpel"/>
  <property name="bpel.preference.isExecutingInitODRequestReceivedCallout">true</property>
  <property name="bpel.preference.isExecutingInitODEventRequestCompleted">true</property>
  <property name="bpel.preference.isExecutingODEventNotificationArrived">true</property>
  <property name="bpel.preference.timeout.callback.years">0</property>
  <property name="bpel.preference.timeout.callback.months">0</property>
  <property name="bpel.preference.timeout.callback.days">0</property>
  <property name="bpel.preference.timeout.callback.hours">0</property>
  <property name="bpel.preference.timeout.callback.minutes">0</property>
  <property name="bpel.preference.timeout.callback.seconds">45</property>
  <property name="bpel.preference.OD_ServerCallbackEndpoint">http://127.0.0.1:8001/soa-infra/serv
  <property name="bpel.preference.externalSenderID">DSC Sensus</property>
  <property name="partnerLink.OD_ServerProxy.idempotent" type="xs:string"
            many="false">false</property>
</component>
```

Properties must be prefixed with “bpel.preference.” The BPEL process associated with them (see the implementation.bpel element) can use the ora:getPreference() method to extract the value. When accessing the values, the “bpel.preference” prefix is dropped.

### Pre-Deployment Changes

While the values of properties cannot be changed at runtime, they can be altered using the configuration plan each composite uses for deployment. Most configuration plans will contain multiple property elements within the “component” section. Modifying the value in the replace element and redeploying the composite will change the behavior of the property:

```
<component name="DeviceStatusCheck">
   <property name="bpel.preference.isExecutingInitODRequestReceivedCallout">
      <replace>true</replace>
   </property>
   <property name="bpel.preference.isExecutingInitODEventRequestCompleted">
      <replace>true</replace>
   </property>
   <property name="bpel.preference.isExecutingODEventNotificationArrived">
      <replace>true</replace>
   </property>
   <property name="bpel.preference.timeout.callback.years">
      <replace>0</replace>
   </property>
   <property name="bpel.preference.timeout.callback.months">
      <replace>0</replace>
   </property>
   <property name="bpel.preference.timeout.callback.days">
      <replace>0</replace>
   </property>
```

When the composite is deployed, the properties will contain the values in the “replace” elements.


Copyright © 2026, Oracle and/or its affiliates.

#### Post-Deployment Changes

After deployment, changes can still be made to the values of the properties through Enterprise Manager.

1. In the Enterprise Manager navigation window, open WebLogic Domain/{domain name}. Right click on the domain and select **System MBean Browser**.
2. A new navigation pane opens in the window on the right. Select **Application Defined MBeans/oracle.soa.config/Server: {domain name}/SCAComposite/{Composite} [1.0]/ SCAComposite.SCAComponent/{Component}** to access the controls for a particular component. A “component” is a single BPEL or Mediator within a larger SOA composite

   A list of options will open in the right pane.
3. Select **Properties.**

   The Property sheet shows a list of all the properties. Opening one allows editing of the property value.
4. Click **Apply** to save any changes.

### Proxies

Proxy web services make no changes to the data going through the system and are just used to centralize endpoint configuration. The message structures are identical to those of the head-end system. Data is passed through without modification. Mediator objects within the Common composite are used to move data from Oracle Utilities Smart Grid Gateway to the head-end system.

### Process Callouts

Some services are designated as “Process Callouts.” These services allow integrators direct access to the XML coming to and from the head-end system. They are useful in the event the integrator needs to provide custom enrichment or transformation of the inbound or outbound data. Because process callouts are generally located before and after calls to the head-end system or when data arrives, they can also serve as an “event” trigger if the integrator requires additional processing logic. The services define identical inputs and outputs and are based on head-end schema definitions. To use a callout, the integrator should create a web service implementing one or more of the WSDLs defined in the Common composite. Other composites should then be configured to target the new Callout-type service.

A typical example is for re-rendering the meter identifier being sent to the head-end system. If the standard meter ID should be modified, a process callout service is a candidate for making this change. Note that because the schema is based off the head-end system's definition, deviations from the schema are not allowed.

## Configuring and Customizing Adapter BPEL Processes

This section outlines how to configure and customize the Oracle Business Process Execution Language (BPEL) processes provided with the Adapter Development Kit to work with your implementation.

- Editing Configuration Files
- Customizing BPEL Processes


Copyright © 2026, Oracle and/or its affiliates.

This section outlines the changes that need to made in various configuration and build files to enable the application servers used in your implementation to communicate with each other, and to support the specifics of your head-end system. These files are also used in packaging and deploying the processes.

**Server Definitions**

In order for your adapter to work properly, you must set up the application servers that run the components used by the adapter communicate with each other as follows:

- The Oracle Utilities Smart Grid Gateway application server must be able to send and receive messages to and from the SOA Suite application server.
- The SOA Suite application server must be able to send and receive messages to and from the head-end system application server, and the Oracle Utilities Smart Grid Gateway application server
- The head-end system application server must be able to receive and send messages from and to the SOA Suite application server.

The types of servers and the tokens that need to be replaced for each are listed below. Further below are the specific configuration files that must be modified, as well as the locations in each files that must be modified. Note that a port is listed, but may not be necessary depending on the type of installation. Also note that the SOA Server will have to have the partition name defined that is used.

- **SOA Server**: This is the application server running the SOA Suite components, including Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL). The SOA server is referenced using the following tokens:
  - {SOA_HOST}: the server where the SOA server has been installed
  - {SOA_PORT_NUMBER}: the port used by the SOA server
  - {SOA_PARTITION_DG}: the partition used by the SOA server. There are different partitions for the different adapters used by Oracle Utilities Smart Grid Gateway.
- **XAI (OUAF) Server**: This is the application server running the Oracle Utilities Smart Grid Gateway (including the Oracle Utilities Application Framework, or OUAF) software, including the inbound/outbound message components. This server is referenced using the following tokens:
  - {WEB_WLHOST}: the server where Oracle Utilities Smart Grid Gateway and the message components have been installed
  - {WEB_WLPORT}: the port used by the Oracle Utilities Smart Grid Gateway application
  - {WEB_CONTEXT_ROOT}: the root directory at which the Oracle Utilities Smart Grid Gateway inbound web service
- AMI Head-end Server: This is the application server running the head-end system software. The AMI head-end server is referenced using the following tokens:
  - {Headend_MR_Server_DG}: the URL for either the actual head-end system's MR_Server MultiSpeak implementation, or the URL to an emulator test harness being used (if applicable). For example, the Adapter Development Kit includes a soapUI configuration that can be used to emulate a head-end system for testing purposes.
  - {Headend_CD_Server_DG}: the URL for either the actual head-end system's CD_Server MultiSpeak implementation, or the URL to an emulator test harness being


Copyright © 2026, Oracle and/or its affiliates.

used (if applicable). For example, the generic adapter includes a soapUI configuration that can be used to emulate a head-end system for testing purposes.

  - {Headend_OD_Server_DG}: the URL for either the actual head-end system's OD_Server MultiSpeak implementation, or the URL to an emulator test harness being used (if applicable). For example, the generic adapter includes a soapUI configuration that can be used to emulate a head-end system for testing purposes.

In addition, credentials for the WebLogic server must be specified in the build.properties file in the deploy folder:

- {WebLogic_UserID}: the user ID used to connect to the WebLogic server
- {WebLogic_Password}: the password for the {WebLogic_UserID} used to connect to the WebLogic server

## Customizing BPEL Processes

This section provides an overview of how the sample BPEL processes provided with the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be customized to meet specific business requirements, such as using custom XSL transformations and adding/editing the steps involved in the process.

### Using Custom XSL Files

This process references XSL files to transform the messages from the Oracle Utilities Smart Gird Gateway standard format into the format used by the “generic” head-end system (based on the Multispeak protocol) when sending messages to the head-end system, and from the head-end system format into the Oracle Utilities Smart Gird Gateway standard format when sending messages back to Oracle Utilities Smart Gird Gateway. These XSL file references can be changed to custom XSL files that transform the messages into and from the format used by your head-end system. Refer to Oracle BPEL and Oracle JDeveloper documentation for more information about referencing XSL files within a BPEL process.

### Making Changes to the Process

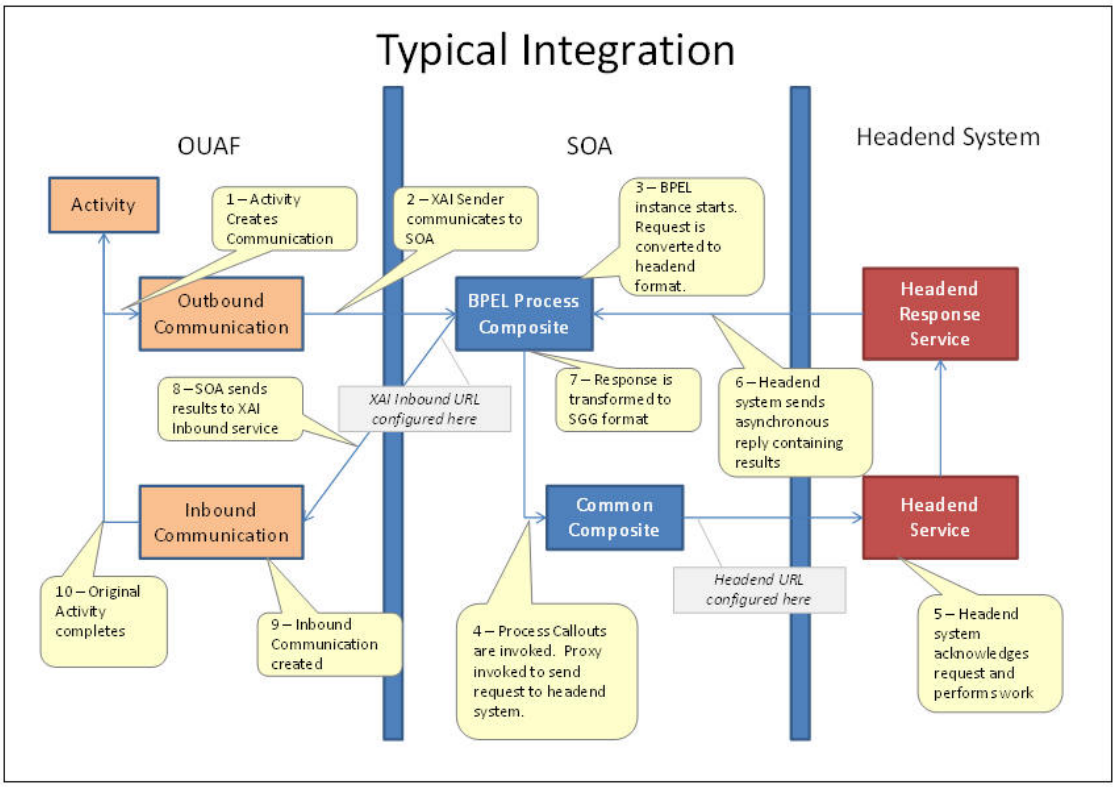
In addition to referencing custom XSL files for transformation, you can also add additional steps to the process to meet the specific business requirements of your implementation, or make other changes to the existing steps. Refer to Oracle BPEL and Oracle JDeveloper documentation for more information about adding and/or changing steps within BPEL process.

Note: Any changes made to this process for a particular AMI vendor and head-end system needs to be modeled based on the head-end system/AMI vendor’s business processes, including configuring the required request/response transactions required for a specific command (such a remote connect).

# Commands

This section describes the general structure and flow of each SOA composite in the adapter development kit for the two-way commands. Most follow this basic model:


Copyright © 2026, Oracle and/or its affiliates.




- Common
- Commission / Decommission
- Connect / Disconnect
- Device Status Check
- On Demand Read

## Common

The Common composite is a repository for files and logic that is used by other composites. For instance, the Proxies and default implementations of Process Callouts are defined in the Common composite.

### Composite Properties

None.

### Composite WSDLs

### Callout Services

Process Callouts are customization points for integrators with identical inputs and outputs based on head-end formats. The Common composite contains default implementations in which incoming data is reflected back in an identical state. Each of the following is a Mediator component and uses an “echo”-type definition.

Process Callouts are usually associated with boolean composite properties which control whether they will be executed. They are activated by default so uses can easily inject their customizations, but setting the properties to false can be a performance optimization. When


Copyright © 2026, Oracle and/or its affiliates.

set to false, the associated Process Callout will not be made and execution will continue normally.

| Service | Description |
|---|---|
| OA_CalloutService | Used when the asynchronous reply in DeviceStatusCheck arrives. |
| MR_CalloutService | Used during Commission/Decommission and OnDemandRead operations. |
| CB_CalloutService | Used by the asynchronous callback processes of OnDemandRead and Connect/Disconnect. |
| CD_CalloutService | Used by Connect/Disconnect. |
| OD_CalloutService | Used in DeviceStatusCheck. |

### Proxies

Placing all the proxy web services in the Common composite creates a place to conveniently set endpoints. Each is a Mediator component which passes the data to the head-end system without modification

| Proxy Web Service | Description |
|---|---|
| OD_ServerProxy | Used to transmit a DeviceStatusCheck request to the head-end system. |
| CD_ServerProxy | Used to transmit a Connect/Disconnect request to the head-end system. |
| MR_ServerProxy | Used to transmit an OnDemandRead request to the head-end system. |

### Other

**AuxiliaryRoutinesService:** A container defining helpful, commonly used functions:

- **FindExpTime:** Many MultiSpeak functions contain an expTime element, which is used to deliver the length of time to wait for the command to complete before failing. (The included test harness does not support timeouts.) If the input to the command composite does not contain this timeout, the process will read from its properties the years, months, days, hours, minutes, and seconds to wait for a result. This web service utilizes a Java method to combine these inputs into the proper XML field used in the MultiSpeak API. The properties are also used to control the timeout wait period of asynchronous callbacks within a command's BPEL process.
- **FindTimeout:** When the expTime element and units are supplied as input, it is usually still necessary to compute the timeout used in asynchronous callbacks. This method accepts the MultiSpeak fields and replies in the “P0Y0M0DT0H0M0S” format understood by BPEL.

## Commission / Decommission

The CommissionDecommission composite takes care of registering and de-registering a device with the head-end system.


Copyright © 2026, Oracle and/or its affiliates.

### Composite Properties

| Property Name | Default Value | Description |
|---|---|---|
| isExecutingCommissionReceived-Callout | true | Controls whether the Request Arrived callout executes. |
| isExecutingCommissionCompleted-Callout | true | Controls whether the Request Completed callout executes. |
| isExecutingDecommissionReceived-Callout | true | Controls whether the Request Arrived callout executes. |
| isExecutingDecommissionCompleted-Callout | true | Controls whether the Request Completed callout executes. |

### Composite WSDLs

| Composite | Description |
|---|---|
| CommissionDecommission Service | Entry point for the CommissionDecommission command. The operations are synchronous, so no additional WSDLs are needed. |
| MR_ServerCallout | Reference describing the process callout. |
| MR_ServerProxy | Reference describing the proxy used to invoke the head-end system. |

### Process Flow

1. SGG/OUAF invokes MeterAddNotification/MeterRemoveNotification operation using CommissionDecommissionService.
2. Test whether MeterAddNotification or MeterRemoveNotification is invoked. In both cases, the following steps are common for both the operations.
3. Composite properties are loaded and local variables are initialized.
4. Both the Header and Body inputs are transformed to MultiSpeak format.
5. If required, execute a process callout in MR_ServerCallout.Assign updated data to head-end request.
6. Invoke MeterAddNotification/MeterRemoveNotification via the MR_ServerProxy to the head-end system.
7. If required, execute a process callout in MR_ServerCallout. Assign updated data to head-end response.
8. Transform Header and Body head-end responses to SGG/OUAF format.
9. Reply to SGG/OUAF with synchronous results.

## Connect / Disconnect

The ConnectDisconnect composite is responsible for starting and stopping the recording of usage data for a meter.


Copyright © 2026, Oracle and/or its affiliates.

### Composite Properties

| Property Name | Default Value | Description |
|---|---|---|
| CB_CDCallbackEndpoint | http://[SERVER_NAME]/soa-infra/services/DG/ConnectDisconnect/CB_ServerService | The default endpoint to which MultiSpeak should send the asynchronous CB_Server callback. |
| isExecutingCDReceivedCallout | true | Controls whether the Request Received callout executes. |
| isExecutingCDCompletedCallout | true | Controls whether the Request Received Completed callout executes. |
| isExecutingCDStatesChangedArrivedCallout | true | Controls whether the CD States Changed Notification arrival callout executes. |
| CallbackTimeoutYears | 0 | The number of years to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutMonths | 0 | The number of months to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutDays | 0 | The number of days to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutHours | 0 | The number of hours to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |


Copyright © 2026, Oracle and/or its affiliates.

| Property Name | Default Value | Description |
|---|---|---|
| CallbackTimeoutMinutes | 0 | The number of minutes to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutSeconds | 45 | The number of seconds to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |

### Composite WSDLs

| Composite | Description |
|---|---|
| ConnectDisconnectService | Entry point for the ConnectDisconnect operation. After a synchronous reply, further updates will be sent to the specified location asynchronously using DG-ConnectDisconnectStateChgNotification. |
| CB_ServerService | MultiSpeak WSDL hosted to receive asynchronous CDStatesChangedNotification callbacks from the head end system. |
| CB_ServerCallout | Reference describing the process callout used when the asynchronous callback from the head end system arrives. |
| CD_ServerCallout | Reference describing the process callout. |
| CD_ServerProxy | Reference describing the proxy used to invoke the head end system. |
| AuxiliaryRoutines | Reference to common helper routines in the Common composite. |
| DG-ConnectDisconnectStateChgNotification | A reference to an inbound web service capable of processing asynchronous results from a Connect/Disconnect request. |

### Process Flow

1. SGG/OUAF invokes InitiateConnectDisconnect operation using ConnectDisconnectService.
2. Composite properties are loaded and local variables are initialized.
3. Test whether the input includes a expiration time
   - If Yes: Use AuxilaryRoutines to compute timeout.
   - If No: Use composite properties and AuxiliaryRoutines to compute expiration time.
4. Both the Header and Body inputs are transformed to MultiSpeak format.
5. If required, execute a process callout in CD_ServerCallout. Assign updated data to head end request.


Copyright © 2026, Oracle and/or its affiliates.

6. Add a callback URL to the head end request. This is the endpoint the head end will use to send the results.
7. Invoke InitiateConnectDisconnect via the CD_ServerProxy to the head end system.
8. If required, execute a process callout in CD_ServerCallout. Assign updated data to head end response.
9. Transform Header and Body head end responses to SGG/OUAF format.
10. Reply to SGG/OUAF with synchronous results.
11. Check for ErrorObject in the head end response.
12. If error object not found:

    Await asynchronous response from head-end system for the period of expiration time calculated during start of this flow:

    - Receive CDStatesChangeNotification from CB_ServiceService.
    - Generate reply with no errors for CDStatesChangeNotification
    - Synchronously reply to the head end.
    - If required, execute a process callout in CB_ServerCallout. Assign updated data to the incoming request.
    - Transform Header and Body of incoming request to OUAF inbound web service format described in DG-ConnectDisconnectStateChgNotification.
    - If request contains responseURL
      - Invoke the inbound web service on given responseURL with the transformed incoming request.
    - Else
      - Invoke the inbound web service on definedURL with the transformed incoming request.

## Device Status Check

DeviceStatusCheck is used to determine the health of a meter and to test that the meter can be reached on the network.

### Composite Properties

| Property Name | Default Value | Description |
|---|---|---|
| OD_ServerCallbackEndpoint | http://127.0.0.1:8001/soa-infra/services/DG/DeviceStatusCheck/OA_ServerService | The default endpoint to which MultiSpeak should send the asynchronous OA_Server callback. |
| isExecutingInitODRequest-ReceivedCallout | true | Controls whether the Request Arrived callout executes. |
| isExecutingInitODEventRequest-Completed | true | Controls whether the Request Completed callout executes. |
| isExecutingODEventNotification-Arrived | true | Controls whether the ODEventNotification arrival callout executes. |


Copyright © 2026, Oracle and/or its affiliates.

| Property Name | Default Value | Description |
|---|---|---|
| timeout.callback.years | 0 | The number of years to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| timeout.callback.months | 0 | The number of months to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| timeout.callback.days | 0 | The number of days to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| timeout.callback.hours | 0 | The number of hours to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| timeout.callback.minutes | 0 | The number of minutes to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| timeout.callback.seconds | 45 | The number of seconds to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |

### Composite WSDLs

| Composite | Description |
|---|---|
| OD_ServerService | Entry point for DeviceStatusCheck. After a synchronous reply, further updates will be sent to the specified location asynchronously using DG-OutageDetectionEventNotification. |
| OA_ServerService | MultiSpeak WSDL hosted to received asynchronous ODEventNotification callbacks from the head end system. |


Copyright © 2026, Oracle and/or its affiliates.

| Composite | Description |
|---|---|
| OD_ServerProxy | Reference describing the proxy used to invoke the head end system to initiate a Device Status Check. |
| OA_ServerCallout | Reference describing the process callout used when the asynchronous callback from the head end system arrives. |
| OD_ServerCallout | Reference describing the process callout used when initiating a Device Status Check. |
| AuxilaryRoutines | Reference to common helper routines in the Common composite. |
| DG-OutageDetectionEventNotificationXAI | A reference to an inbound web service capable of processing asynchronous results from a Device Status Check request. |

### Process Flow

1. SGG/OUAF Initiates the Outage Detection service using OD_ServerService
2. Composite properties are loaded. Local variables are initialized. These mostly include boolean settings describing the state (such as whether or not the synchronous reply from the head-end system returned)
3. Test whether the input includes a expiration time (expTime element).
   - If Yes: Use AuxilaryRoutines to compute timeout
   - If No: Use composite properties and AuxiliaryRoutines to compute expiration time
4. Both the Header and Body inputs are transformed to MultiSpeak format. This can be done either through direct mapping or through XSLT. In this instance, XSLT is used to transform the header and the body separately.
5. If the composite property indicates the process callout should occur, execute the “request arrived” process callout in OD_CalloutService. Assign updated data to head end request.
6. Add a callback URL to the head end request. This is the endpoint the head end will use to send the outage detection results. Some head end systems use other approaches, such as pre-configuring the URL for callbacks or allowing polling to find the results. In this case, the URL is configured in a composite property and corresponds to an implementation of the OA_Server WSDL. When the head end system has results, it will deliver them to this URL using the ODEventNotification method.
7. Invoke InitiateOutageDetectionEventRequest via the OD_ServerProxy to the head end system.
8. Split processing to handle simultaneous activities. The Flow activity does this in BPEL and it makes sense here because one process will handle a synchronous response and return it to the caller. The other process will sleep, or “dehydrate,” until the asynchronous ODEventNotification arrives or until the timeout value is reached.
   - Handle synchronous response to SGG/OUAF
     - If the composite property indicates the process callout should occur, execute the “request completed” process callout in OD_CalloutService. Assign updated data to head end response.
     - Transform Header and Body head end responses to SGG/OUAF format.
     - Reply to SGG/OUAF with synchronous results.


Copyright © 2026, Oracle and/or its affiliates.

- Set boolean variable indicating that the synchronous reply has been returned.

- Await asynchronous response from head-end system.
  - Receive ODEventNotification from OA_ServerService. If the timeout period elapses, raise an error and stop waiting.
  - Synchronously reply to the head end with no errors
  - If the composite property indicates the process callout should occur, execute the “data arrived” process callout in OA_CalloutService. Assign updated data to the incoming request.
  - Transform Header and Body to inbound web service format described in DG-OutageDetectionEventNotification.
  - Invoke the inbound web service with the new data.

9. Fault Handlers look for any remote Fault returned from the head-end system. The MultiSpeak API does not define named faults, but they are still possible to encounter. For example, a security error or network error could be raised.
   - If the synchronous reply has been sent back to the caller, Terminate the flow. This indicates an error has occurred and more research in Enterprise Manager is warranted.
   - If the synchronous reply has not been sent, stop further processing and compose a reply which contains the fault information.

## On Demand Read

OnDemandRead interrogates a meter for the usage at the current point in time.

**Composite Properties**

| Property Name | Default Value | Description |
|---|---|---|
| CB_ServerCallbackEndpoint | http:/[SERVER_NAME]/soa-infra/services/DG/OnDemandRead/CB_Server | The default endpoint to which MultiSpeak should send the asynchronous CB_Server callback |
| IsExecutingOnDemandReadRequestReceived | true | Controls whether the Request Received callout executes |
| IsExecutingOnDemandReadRequestReceivedResponse | true | Controls whether the Request Received Completed callout executes |
| IsExecutingReadingChangedNotification | true | Controls whether the Reading Changed Notification arrival callout executes |
| CallbackTimeoutYears | 0 | The number of years to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |


Copyright © 2026, Oracle and/or its affiliates.

| Property Name | Default Value | Description |
|---|---|---|
| CallbackTimeoutMonths | 0 | The number of months to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutDays | 0 | The number of days to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutHours | 0 | The number of hours to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutMinutes | 0 | The number of minutes to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |
| CallbackTimeoutSeconds | 45 | The number of seconds to await a response from the MultiSpeak server. This property only takes effect when the expTime element is not supplied in the input to the service. |

### Composite WSDLs

| Composite | Description |
|---|---|
| OnDemandReadService | Entry point for the OnDemandRead operation. After a synchronous reply, further updates will be sent to the specified location asynchronously using DG-ReadingChangedNotification. |
| CB_ServerService | MultiSpeak WSDL hosted to receive asynchronous ReadingChangedNotification callbacks from the head end system. |
| MR_ServerProxy | Reference describing the proxy used to invoke the head end system to initiate a On Demand Read. |


Copyright © 2026, Oracle and/or its affiliates.

| Composite | Description |
|---|---|
| MR_ServerCallout | Reference describing the process callout used when initiating a On Demand Read. |
| CB_ServerCallout | Reference describing the process callout used when the asynchronous callback from the head end system arrives. |
| AuxilaryRoutines | Reference to common helper routines in the Common composite. |
| DG-ReadingChangedNotificationXAI | A reference to an inbound web service capable of processing asynchronous results from a On Demand Read request. |

### Process Flow

1. SGG/OUAF invokes InitiateMeterReadingsByMeterID operation using OnDemandReadService.
2. Composite properties are loaded. Local variables are initialized.
3. Test whether the input includes a expiration time
   - If Yes: Use AuxilaryRoutines to compute timeout
   - If No: Use composite properties and AuxiliaryRoutines to compute expiration time
4. Both the Header and Body inputs are transformed to MultiSpeak format.
5. If required, execute a process callout in MR_ServerCallout. Assign updated data to head end request.
6. Add a callback URL to the head end request. This is the endpoint the head end will use to send the results.
7. Invoke InitiateMeterReadingsByMeterID via the MR_ServerProxy to the head end system.
8. If required, execute a process callout in MR_ServerCallout. Assign updated data to head end response.
9. Transform Header and Body head end responses to SGG/OUAF format
10. Reply to SGG/OUAF with synchronous results
11. Check for ErrorObject in the head end response.
12. If error object not found
    - Await asynchronous response from head end for the period of expiration time calculated during start of this flow
      - Receive ReadingChangedNotification from CB_ServiceService.
      - Generate reply with no errors for ReadingChangedNotification.
      - Synchronously reply to the head end.
      - If required, execute a process callout in CB_ServerCallout. Assign updated data to the incoming request.
      - Transform Header and Body of incoming request to OUAF inbound web service format described in DG-ReadingChangedNotificationXAI.
      - If request contains responseURL
        * Invoke the inbound web service on given responseURL with the transformed incoming request.


Copyright © 2026, Oracle and/or its affiliates.

- Else
  - Invoke the inbound web service on definedURL with the transformed incoming request.

## Working with Enterprise Manager

Oracle Enterprise Manager (OEM) is useful for troubleshooting and diagnosing issues communicating with the head end system. In particular, security and other communications issues can be discerned.

To locate the instance of a service that is showing a problem, open the **Dashboard** view of the service. Each run instance is time-indexed. The **State** column in the first table contains the most important information for each instance. Completed messages show up with a green check icon. Terminated instances are indicated with a grey stop icon. Instances that are still running are marked as such. Runtime exceptions are in the bottom window.

When you click on the instance ID, a new window opens showing the entire process flow. This view is particularly useful in debugging.

Note the **State** column showing the status of each instance within the flow. SOA processes are made up of several calls to SOA components and web services. The **Instance** column shows a rough ordering of the operations. A typical approach to troubleshooting would find the “lowest” instance of an error or termination. This would be the error that is deepest within the process and is usually the source of the problem. In the above case, the deepest error is in the very first composite for the process. More information can be found by clicking into the link within the **Instance** column.

The level of detail is dependent on the auditing level set on the server. In this case, the error is in the response from the head end system.

OEM also contains a centralized location to control security. On a **Dashboard** screen for a composite, the **Policies** tab shows the OWSM policies attached to the composite. Typically, a log policy is placed on any inbound or outbound communication. Also, all services or references are delivered with attached basic http security policies.

## MultiSpeak Implementation

This section lists the subset of MultiSpeak schemas and WSDLs utilized by the Adapter Development Kit.

**MultiSpeak Head End System Service Definitions**

- CB_Messages.wsdl
- CB_Server.wsdl
- CD_Messages.wsdl
- CD_PortTypes.wsdl
- CD_Server.wsdl
- MR_Messages.wsdl
- MR_PortTypes.wsdl
- MR_Server.wsdl
- OA_Messages.wsdl
- OA_Server.wsdl


Copyright © 2026, Oracle and/or its affiliates.

- OA_ServerCollisions.wsdl
- OD_Messages.wsdl
- OD_PortTypes.wsdl
- OD_Server.wsdl

**MultiSpeak Head End XML Schemas**

- cpsm.xsd
- gml.xsd
- xlink.xsd
- mspcommon.xsd
- CB_Server.xsd
- CD_Server.xsd
- MR_Server.xsd
- OA_Server.xsd
- OD_Server.xsd

# Configuring an Adapter Development Kit Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway to communicate with the smart meters.

- Inbound Web Services
- Message Senders
- Outbound Message Types
- External System
- Service Provider
- Processing Methods
- Configuring Endpoint URIs

## Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the head-end system in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

| Inbound Web Service Name | Description |
|---|---|
| D1-BulkRequestHeader | Bulk Request Header |
| D1-BulkRequestUpdate | Bulk Request Update |
| D1-BulkResponse | Bulk Response |
| D1-DeviceEventSeeder | Device Event Seeder |


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service Name | Description |
|---|---|
| D1-DeviceStatusCheck | Device Status Check |
| D1-InitialLoadIMD | IMD Seeder |
| D1-PayloadErrorNotif | Payload Error Notification |
| D1-PayloadStatistics | Payload Statistics |
| D1-PayloadSummary | Payload Summary |
| DG-ConDisconStChgNotification | Initiate Connect Disconnect Response |
| DG-OutageDetectionEventNotification | Initiate Outage Detection Response |
| DG-ReadingChangedNotification | Reading Changed Notification |

Note: The following apply to all of the above inbound web services:

**Message Options**

- **Trace**: No
- **Debug**: No
- **Active**: Yes

**Operations**

- **Operation Name**: Same as web service name
- **Schema Type**: Business Object
- **Schema Name**: Applicable business object code
- **Transaction Type**: Add

## Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

| Message Sender | Description |
|---|---|
| DG-COMM | Generic Adapter Commission |
| DG-DCOMM | Generic Adapter Decommission |
| DG-INTOUTDET | Initiate Outage Detection Request |
| DGXAISender | Used for Meter Decommission |

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type**: Real-time
- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:


Copyright © 2026, Oracle and/or its affiliates.

- **HTTP Header**: SOAPAction: **<OPERATION>**
- **HTTP Login User**: **<USER_ID>**
- **HTTP Login Password**: **<PASSWORD>**
- **HTTP Method**: POST
- **HTTP URL 1**: http://**<EM_SERVER>**:**<EM_SERVER_PORT>**/soa-infra/services/Generic/**<SERVICE>**

  where:

  - **<OPERATION>**: the operation performed by the message sender (see Operation column in the table above)
  - **<USER_ID>**: the user ID used to log into WebLogic Enterprise Manager
  - **<PASSWORD>**: the password used to log into WebLogic Enterprise Manager
  - **<EM_SERVER_IP>**: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
  - **<EM_SERVER_PORT>**: the port where the WebLogic Enterprise Manager is installed
  - **<SERVICE>**: the service invoked by the message sender (see Service column in the table above)

## Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

| Outbound Message Type | Description |
|---|---|
| DG-COMM | Generic Adapter Commission |
| DG-DCOMM | Generic Adapter Decommission |

Note: The following apply to all of the above outbound message types:

- **Business Object**: D1-OutboundMessage (Outbound Message)
- **Priority**: Priority 50

## External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the head-end system must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

**External System — Generic**:

- **External System**: Generic


Copyright © 2026, Oracle and/or its affiliates.

- **Description**: Generic
- **Outbound Message Types**:

| Outbound Message Type | Description | Message Sender |
|---|---|---|
| DG-COMM | Generic Adapter Commission | DG-COMM |
| DG-DCOMM | Generic Adapter Decommission | DG-DCOMM |

Note: The following apply to all of the above outbound message types:

- **Processing Method**: Real-time
- **Message XSL**: DG-Request xsl
- **Response XSL**: DG-Response.xsl

## Service Provider

Service providers represent external entities that serve various roles relative to the application, including head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the head-end system must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

**Service Provider - Generic**:

- **Service Provider**: Generic
- **Description**: Generic
- **External Reference ID**: Generic
- **External System**: Generic
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **AMI Measuring Component ID Type**: Channel ID

## Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to the head-end system.

The following types of processing methods must be configured for the head-end system service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about configuring processing methods.


Copyright © 2026, Oracle and/or its affiliates.

### Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from the head-end system.

### Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from the head-end system.

### UOM Translation

UOM translation processing methods define how head-end-specific unit of measure (UOM) codes are mapped to standard UOM codes. This processing method is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

### Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the head-end system service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

| Command | Processing Role | Default Business Object | Default Outbound Message Type |
|---|---|---|---|
| Device Commission | Device Registration | DG-MeterAddNotification | Commission |
| Device Decommission | Device Removal | DG-MeterRemoveNotification | Decommission |
| Device Status Check | Device Status Check | DG-InitiateOutageDetection | Get Status |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | DG-InitiateMeterByMeterId | Initiate Meter Read by Meter ID |
| Remote Connect | Remote Connect | DG-InitiateConnectDisconnect | Connect Device |
| Remote Disconnect | Remote Disconnect | DG-InitiateConnectDisconnect | Disconnect Device |

## Configuring Endpoint URIs

Part of configuring your adapter is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands.

The default approach to defining Endpoints URIs is redeployment or reinstallation or the BPEL composites. For example, by default, changing an adapter from using the test harness to a


Copyright © 2026, Oracle and/or its affiliates.

production environment using the actual head end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an "Endpoint Override" Domain Value Map (DVM) to override Endpoints URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Adapter Development Kit endpoint override DVM (ADK–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Landis+Gyr adapter and the command used with each:

| DVM Key | Commands |
|---|---|
| MR_Server | Device Commissioning<br>Device Decommissioning<br>On-Demand Read (Scalar)<br>On-Demand Read (Interval) |
| CD_Server | Remote Connect<br>Remote Disconnect |
| OD_Server | Device Status Check |

To define an override Endpoint URI for an adapter built from the Adapter Development Kit, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.

   The URL for the SOA Composer is http://server:port/soa/composer.

2. Select the ADK–EndpointOverrides.dvm in the left panel.

   - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
   - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

   The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.
4. Click the **Add Domain Values** icon ("+") to add a new key.

   The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

   - **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
   - **EndpointURI**: The override Endpoint URI.

6. Click **OK**.

   The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.
8. Click the **Publish** button to activate all the changes in the editing session.


Copyright © 2026, Oracle and/or its affiliates.

Enter an optional note for the session in the **Publish Session** dialog.

Changes take effect immediately upon publishing the session.

Click **Discard** to discard your changes.

Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

## Configuring Adapter Development Kit Extendable Lookups

> **Note**
>
> This section applies to to **both** middleware and native implementations of the Smart Grid Gateway Adapter Development Kit,

This section outlines some of the extendable lookups that must be configured for use with a customized adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

### Device Event Mapping

The Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the head-end system.

Each value defined for the Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system.
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

### Head-End UOM Code to Standard UOM Mapping

Usage received from the head-end system may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The head-end system UOM Code to Standard UOM Mapping extendable lookup is used to determine how to map head-end system UOM codes to standard UOM codes when receiving usage from the head-end system.

Each value defined for the head-end system UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Head-End Unit of Measure**: The unit of measure code used by the head-end system
- **Unit of Measure**: The unit of measure defined in the system.
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive)

### Interval Status Code to Condition Mapping

Interval usage received from the head-end system can include interval status codes that indicate the status or condition of the interval value. These interval status codes must be


Copyright © 2026, Oracle and/or its affiliates.

mapped to standard condition codes in the system. The Generic Interval Status Code to Condition Mapping extendable lookup is used to determine how to map head-end system interval status codes to standard status codes when receiving usage from the head-end system.

Each value defined for the Generic Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status**: The head-end system interval status code
- **Description**: A description of the interval status code.
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Condition**: The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

**Other Extendable Lookups**

| Business Object Name | Description |
|---|---|
| DG-CDReasonCodeLookup | ADK - Connect Disconnect Reason Code |
| DG-GenericTimeUnits | ADK - Time Units |
| DG-LoadActionCodeLookup | ADK - Load Action Code |
| DG-OutageEventTypeLookup | ADK - Outage Event Type |
| DG-ServiceTypeMappingLookup | ADK - Service Type Mapping |

# Using the Adapter Development Kit Test Harness

Oracle Utilities Smart Grid Gateway Adapter Development Kit includes a test harness that can be configured to simulate a general head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. This section describes the test harness and its components.


- Locating the WSDL for the Test Harness
- Web Services
- Sample Meters File


## Locating the WSDL for the Test Harness

Follow these procedures to locate the Adapter Development Kit test harness WSDL:

**How to Use Enterprise Manager to Locate the WSDL**

1. Open Enterprise Manager and use the navigation pane to open the dashboard of the test harness composite:
2. The top bar of the dashboard contains several buttons and icons. One of these is a “world” icon with a puzzle piece over it. Click this icon to display a list of the WSDLs and endpoint URIs for the composite:
3. Click the UtilService WSDL URL link to see the WSDL in the browser, or right click and save it to your machine

Depending on your requirements, it may be necessary to download the associated schema found in the wsdl:types section. The URL can be pasted into a browser tab and downloaded in



Copyright © 2026, Oracle and/or its affiliates.

the same manner as the WSDL. The main schema has imported schemas that may also be required.

**How to Use a Direct URL to locate the WSDL**

The WSDL can be accessed without Enterprise Manager by understanding the paths used on the SOA server. In general, they have the following form:

```
http://{server name}:{port number}/soa-infra/services/{partition}/{Composite}/
{Web Service}?WSDL
```

So by default, the test harness WSDL can be found at

```
http://{server name}:{port number}/soa-infra/services/DG_Test/DGTestHarness/
UtilService?WSDL
```

## Web Services

This section describes the web services included in the Adapter Development Kit test harness BPEL composite.

- General Services
- Locate Meter Services
- Meter Administration Services
- Meter Attribute Administration Services

## General Services

This section describes the general services of the Adapter Development Kit test harness composite.

**LoadMeterIndex**

This web service loads the data store from the internal file. By default if the store is already in memory, it will *not* reload. This behavior can be overridden with the forceReload parameter.

**Input:** LoadMeterIndexInput

**Part:** payload

**Element:** LoadMeterIndexRequest

| Parameter | Description |
|---|---|
| forceReload | A switch telling the system whether to reload the meter index from the configuration file. Default value is false. |

**Output:** LoadMeterIndexOutput

**Part:** payload

**Element:** LoadMeterIndexResult


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Description |
|---|---|
| loaded | A boolean value for whether or not the index was reloaded from the configuration file |

**Fault**: UtilityFault (see UtilityFault for more details).

#### ViewAuditTrail

This web service returns the audit log for the entire session.

**Input:** ViewAuditTrailInput

**Part:** payload

**Element:** ViewAuditTrailRequest

**Parameters:** This is an empty request. There are no parameters.

**Output:** ViewAuditTrailOutput

**Part:** payload

**Element:** ViewAuditTrailResult

This element is an entry consisting of a timestamp and an Operation. Each entry may have an associated meter object showing what changed.

**Fault**: UtilityFault (see UtilityFault for more details).

#### UtilityFault

Fault with similar mapping to SGG/OUAF faults:

Typically, the faultCode, faultString, faultActor, and detail/text elements will be populated.

## Locate Meter Services

This section describes the locate meter web services of the Adapter Development Kit test harness composite.

#### FindMeters

This web service queries the data store for one or more meters. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

**Input:** FindMetersInput

**Part:** payload

**Element:** FindMetersRequest

| Parameter | Description |
|---|---|
| id | The meter ID for which to search |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Description |
|---|---|
| isRegex | The provided id can be a regex value when this parameter is true. Hint: to search for all meters in the system, use ".*" for the ID. |

**Output:** FindMetersOutput

**Part:** payload

**Element:** FindMetersResult

Zero or more meter objects can be returned from the search

**Fault:** See UtilityFault. Unlike other methods, FindMeters does not throw an exception if the meter is not found.

### IsMeterDefined

This web service queries whether a particular meter is defined in the data store.

**Input:** IsMeterDefinedInput

**Part:** payload

**Element:** IsMeterDefinedRequest

| Parameter | Description |
|---|---|
| id | The meter ID for which to search |

**Output:** IsMeterDefinedOutput

**Part:** payload

**Element:** IsMeterDefinedResult

Whether or not the provided ID is part of the index.

**Fault:** See UtilityFault. Thrown when meter id is not found.

### GetMeter

This web service returns all the attributes of a single meter from the in-memory data store. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

**Input:** GetMeterInput

**Part:** payload

**Element:** GetMeterRequest

| Parameter | Description |
|---|---|
| id | The meter ID for which to search |

**Output:** GetMeterOutput


Copyright © 2026, Oracle and/or its affiliates.

**Part:** payload

**Element:** GetMeterResult

The meter object requested by the ID.

**Fault:** See UtilityFault. Thrown when meter id is not found.

## Meter Administration Services

This section describes the meter administration services of the Adapter Development Kit test harness composite.

### AddMeters

This web service adds a set of meters to the in-memory data store. This will not permanently add it to the control file.

**Input:** AddMetersInput

**Part:** payload

**Element:** AddMetersRequest

| Parameter | Description |
|---|---|
| id | The identification code for the meter. |
| utility | An informational string. |
| serviceType | One of the valid ServiceType values (see schema). "Electric" is the only option at this time. |
| isCommissioned | Whether or not the meter is in a commissioned state. |
| loadActionCode | One of the possible LoadActionCode values used in Connect and Disconnect (see schema). |
| outageEventType | One of the possible OutageEventType values used in Device Status Check (see schema). |
| executionStatus | One of the possible ExecutionStates (see schema). These values control how the meter will respond to commands. |
| updateIfExisting | Whether or not to update the meter with the provided values if it already exists in the index. |
| Comment | An informational string describing the purpose of the meter. |
| Channels | A listing of unit of measures supported by this meter. |
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |
| decimals | The number of digits to the right of the decimal that should be generated when reading the meter. |
| description | A longer description of the unit of measure. |

**Output:** AddMetersOutput

**Part:** payload

**Element:** AddMetersResult

Whether or not each meter was added to the index.

**Fault:** See UtilityFault.


Copyright © 2026, Oracle and/or its affiliates.

### RemoveMeter

This web service removes a meter from the in-memory data store. This will not permanently remove it from the control file.

**Input:** RemoveMeterInput

**Part:** payload

**Element:** RemoveMeterRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter to be removed. |

**Output:** RemoveMeterOutput

**Part:** payload

**Element:** RemoveMeterResult

Whether or not the meter was removed from the index.

**Fault:** See UtilityFault. Thrown when meter id is not found.

### AddMeterChannel

This web service adds a new channel to a single meter.

**Input:** AddMeterChannelInput

**Part:** payload

**Element:** AddMeterChannelRequest

| Parameter | Description |
|---|---|
| id | The identification code for the meter. |
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |
| decimals | The number of digits to the right of the decimal that should be generated when reading the meter. |
| description | A longer description of the unit of measure. |

**Output:** AddMeterChannelOutput

**Part:** payload

**Element:** AddMeterChannelResult

Whether or not the channel was added to the index.

**Fault:** See UtilityFault. Thrown when meter id is not found.

### RemoveMeterChannel

This web service removes a Channel from a meter.

**Input:** RemoveMeterChannelInput


Copyright © 2026, Oracle and/or its affiliates.

**Part:** payload

**Element:** RemoveMeterChannelRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter to be removed. |
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |

These three parameters are combined to locate a unique channel

**Output:** RemoveMeterChannelOutput

**Part:** payload

**Element:** RemoveMeterChannelResult

Whether or not the channel was removed from the meter.

**Fault:** See UtilityFault. Thrown when meter id is not found.

#### ReadScalarMeter

This web service generates a scalar reading for each channel of a given meter.

**Input:** ReadScalarMeterInput

**Part:** payload

**Element:** ReadScalarMeterRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter to be read. |

**Output:** ReadScalarMeterOutput

**Part:** payload

**Element:** ReadScalarMeterResult

Zero or more scalar readings for the given meter.

| Parameter | Description |
|---|---|
| uomCode | A code describing the unit of measure for the channel. |
| uomName | A short string containing the name of the unit of measure. |
| decimals | The number of digits to the right of the decimal that should be generated when reading the meter. |
| description | A longer description of the unit of measure. |
| value | A random number representing the scalar reading. |

**Fault:** See UtilityFault. Thrown when meter id is not found.

## Meter Attribute Administration Services


Copyright © 2026, Oracle and/or its affiliates.

This section describes the meter administration services of the Adapter Development Kit test harness composite.

### GetOutageEventType

This web service queries the outage event type for a given meter. The OutageEventType is used by DeviceStatusCheck.

**Input:** GetOutageEventTypeInput

**Part:** payload

**Element:** GetOutageEventTypeRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the OutageEventType should be retrieved. |

**Output:** GetOutageEventTypeOutput

**Part:** payload

**Element:** GetOutageEventTypeResult

The value of the OutageEventType attribute for the requested meter.

**Fault:** See UtilityFault. Thrown when meter id is not found.

### SetOutageEventType

This web service updates the outage event type for a given meter.

**Input:** SetOutageEventTypeInput

**Part:** payload

**Element:** SetOutageEventTypeRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the OutageEventType should be set. |
| value | The new value of OutageEventType to set on the meter. |

**Output:** SetOutageEventTypeOutput

**Part:** payload

**Element:** SetOutageEventTypeResult

The boolean response indicates the success or failure of the update (not the current field status).

**Fault:** See UtilityFault. Thrown when meter id is not found.

### GetLoadActionCode

This web service queries the load action code for a given meter. This is the Connect/Disconnect behavior.

**Input:** GetLoadActionCodeInput


Copyright © 2026, Oracle and/or its affiliates.

**Part:** payload

**Element:** GetLoadActionCodeRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the LoadActionCode should be retrieved. |

**Output:** GetLoadActionCodeOutput

**Part:** payload

**Element:** GetLoadActionCodeResult

The value of the LoadActionCode attribute for the requested meter.

**Fault:** See UtilityFault. Thrown when meter id is not found.

### SetLoadActionCode

This web service updates the load action code for a given meter.

**Input:** SetLoadActionCodeInput

**Part:** payload

**Element:** SetLoadActionCodeRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the LoadActionCode should be set. |
| value | The new value of LoadActionCode to set on the meter. |

**Output:** SetLoadActionCodeOutput

**Part:** payload

**Element:** SetLoadActionCodeResult

The boolean response indicates the success or failure of the update (not the current field status).

**Fault:** See UtilityFault. Thrown when meter id is not found.

### IsCommissioned

This web service queries the commissioning status for a given meter. This is the Commission/ Decommission behavior.

**Input:** IsCommissionedInput

**Part:** payload

**Element:** IsCommissionedRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the Commissioned status should be retrieved. |

**Output:** IsCommissionedOutput


Copyright © 2026, Oracle and/or its affiliates.

**Part:** payload

**Element:** IsCommissionedResult

The value of the Commissioned status attribute for the requested meter.

**Fault:** See UtilityFault. Thrown when meter id is not found.

### SetCommission

This web service updates the commissioning status for a given meter.

**Input:** SetCommissionedInput

**Part:** payload

**Element:** SetCommissionedRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the Commissioned status should be set. |
| value | The new value of Commissioned status to set on the meter. |

**Output:** SetCommissionedOutput

**Part:** payload

**Element:** SetCommissionedResult

The boolean response indicates the success or failure of the update (not the current field status).

**Fault:** See UtilityFault. Thrown when meter id is not found.

### GetExecutionStatus

This web service queries the status of the property controlling the overall execution of the command.

**Input:** GetExecutionStatusInput

**Part:** payload

**Element:** GetExecutionStatusRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the ExecutionStatus should be retrieved. |

**Output:** GetExecutionStatusOutput

**Part:** payload

**Element:** GetExecutionStatusResult

The value of the ExecutionStatus attribute for the requested meter.

**Fault:** See UtilityFault. Thrown when meter id is not found.


Copyright © 2026, Oracle and/or its affiliates.

### SetExecutionStatus

This web service updates the property controlling the overall completion of the command.

**Input:** SetExecutionStatusInput

**Part:** payload

**Element:** SetExecutionStatusRequest

| Parameter | Description |
|---|---|
| id | The ID for the meter for which the ExecutionStatus should be set. |
| value | The new value of ExecutionStatus to set on the meter. |

**Output:** SetExecutionStatusOutput

**Part:** payload

**Element:** SetExecutionStatusResult

The boolean response indicates the success or failure of the update (not the current field status).

**Fault:** See UtilityFault. Thrown when meter id is not found.

## Sample Meters File

The Adapter Development Kit includes an XML file and schema that can be used for configuring one or more meters for use with the test harness. The file, metersdb.xml, is located in the Test/DGTestHarness directory and can be edited with an appropriate XML editor such as XML Spy. This section describes the attributes in the metersdb.xml file.

The meterdb.xml file contains one or more **Meter** elements that have the following attributes:

| Meter Attribute | Definition |
|---|---|
| id | The meter identifier. This value should match the amiDeviceID setting in the Oracle Utilities Application Framework. |
| utility | This is informational only. |
| serviceType | This is informational only. |
| ServiceType | An enumeration that is described in the schema. |
| isCommissioned | A boolean value that describes whether or not the meter is commissioned or decommissioned. The associated MultiSpeak commands are MeterAddNotification and MeterRemoveNotification. |
| loadActionCode | This value is for the Connect and Disconnect commands and has enumeration values described in the schema. The value is returned in the CDStatesChangedNotification MultiSpeak command. |
| outageEventType | This value is used by DeviceStatusCheck and gives the status returned by the ODEventNotification call. Its enumeration values are described in the schema. |


Copyright © 2026, Oracle and/or its affiliates.

| Meter Attribute | Definition |
|---|---|
| executionStatus | This value does not reflect a MultiSpeak command, but is instead intended to give a state of the operation. There are four valid values:<br>**Success:** When a meter has this status the operation will complete without error.<br>**ResponseTimeout:** When a meter has this status an asynchronous reply will never arrive (not relevant for Commission/Decommission commands).<br>**SyncOperationFailure:** When a meter has this status the initial communication to the simulated head-end system will produce an error.<br>**AsyncOperationFailure:** When a meter has this status, the asynchronous callback from the head-end system will arrive, but will indicate an error (not relevant for Commission/Decommission commands). |

A meter can also contain the following elements:

- **Comment:** A field which is for informational use only and is meant to indicate the purpose of the meter.
- **Channels:** Used in reading the meter for On Demand Read commands. A channel contains the following attributes:

| Channel Attribute | Definition |
|---|---|
| uomCode | The units of measure code that should be returned when the meter is read. |
| uomName | The units of measure name that should be returned when the meter is read. |
| description | A longer form. When the meter is read, a random number is generated for this value. |
| decimals | A value to indicate how many places to the right of the decimal that random number should have. |

## The Adapter Development Kit Native Format

> **Note**
>
> The Adapter Development Kit Native Format can be used with **both** middleware and native Smart Grid Gateway Adapter implementations.

The Adapter Development Kit supports loading usage and event data exported from the AMI head-end system in the “native” initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects). Processing of the ADK native format is supported by the following OSB projects:

- **SGG-DG-SEEDER-BASE**
- **SGG-DG-SEEDER-CM**


Copyright © 2026, Oracle and/or its affiliates.

Refer to Initial Measurements and Device Events and OSB Project Summary for more information about these OSB projects.

**Format Details**

The ADK native format is an XML format that contains zero or more initial measurements and/or device events, as follows:

- The collection of initial measurements and/or device events are encapsulated within in an <SGGIMDsEvents> element.
- Each initial measurement or device event is defined in the format of the initial measurement and device event seeder business objects, encapsulated by the following elements:
  - **Initial Measurements**: <D1-InitialLoadIMD>
  - **Device Events**: <D1–DeviceEventSeeder>
- The format can support any number of initial measurements and/or device events.
- Initial measurements can be either scalar or interval measurements.

See Adapter Development Kit Native Format Example for an example of this format. See Adapter Development Kit Native Format Schema for the native format XML schema.

## Adapter Development Kit Native Format Example

The following is an example of the ADK native format. This example contains 2 initial measurements (1 scalar and 1 interval) and 2 device events (PowerOutage and PowerRestored).

```
<?xml version="1.0" encoding="UTF-8" ?>
<SGGIMDsEvents xmlns="http://oracle.com/SGGIMDsEvents">
  <D1-InitialLoadIMD dateTimeTagFormat="xsd" xmlns="">
    <preVEE>
      <imdType>D1IL</imdType>
      <mcIS>D1SC</mcIS>
      <externalId>da_164_Scalar_withMetadata.xml-2015-09-10-11-19-03-297</
externalId>
      <dvcIdN>DL_06</dvcIdN>
      <externalUOM>KWH</externalUOM>
      <enDt>2005-01-01T00:00:00Z</enDt>
      <enQty>4544</enQty>
    </preVEE>
    <serviceProviderExternalId>Itron</serviceProviderExternalId>
  </D1-InitialLoadIMD>
  <D1-InitialLoadIMD dateTimeTagFormat="xsd" xmlns="">
    <preVEE>
      <imdType>D1IL</imdType>
      <mcIS>D1IN</mcIS>
      <externalId>da_164_Multile_withMetadata_A.xml-2015-09-10-11-19-03-297</
externalId>
      <dvcIdN>ZZ-D-OSB-INT-ITRON-0002</dvcIdN>
      <spi>900</spi>
      <externalUOM>KWH</externalUOM>
      <mcm>725</mcm>
      <stDt>2003-01-01T23:45:00Z</stDt>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      <enDt>2003-01-02T00:00:00Z</enDt>
      <msrs>
        <mL>
          <s>1</s><q>4722</q>
          <sts>
            <stsL>
              <s>1</s>
              <st>CHardwareFailure</st>
            </stsL>
            <stsL>
              <s>2</s>
              <st>COverflow</st>
            </stsL>
          </sts>
        </mL>
      </msrs>
    </preVEE>
    <serviceProviderExternalId>Itron</serviceProviderExternalId>
  </D1-InitialLoadIMD>
  <D1-DeviceEventSeeder dateTimeTagFormat="xsd" xmlns="">
    <externalSenderId>Itron</externalSenderId>
        <deviceIdentifierNumber>ZZ-D-OSB-INT-ITRON-0002</
deviceIdentifierNumber>
    <externalEventName>PowerOutage</externalEventName>
    <eventDateTime>2001-02-02T00:00:00Z</eventDateTime>
    <eventInformation>
      <externalEventIdentifier>2147483642</externalEventIdentifier>
      <externalStatusValue>string</externalStatusValue>
      <externalEventCategory>Communication</externalEventCategory>
    </eventInformation>
  </D1-DeviceEventSeeder>
  <D1-DeviceEventSeeder dateTimeTagFormat="xsd" xmlns="">
    <externalSenderId>Itron</externalSenderId>

<externalSourceIdentifier>da_164_Multile_withMetadata_A.xml-2015-09-10-11-19-0
3-297</externalSourceIdentifier>
    <deviceIdentifierNumber>ZZ-D-OSB-INT-ITRON-0002</deviceIdentifierNumber>
    <externalEventName>PowerRestored</externalEventName>
    <eventDateTime>2001-02-02T01:00:00Z</eventDateTime>
    <eventInformation>
      <externalEventIdentifier>2147483642</externalEventIdentifier>
      <externalStatusValue>string</externalStatusValue>
      <externalEventCategory>Communication</externalEventCategory>
    </eventInformation>
  </D1-DeviceEventSeeder>
</SGGIMDsEvents>
```

## Adapter Development Kit Native Format Schema

The following is the XML schema of the ADK native format.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
```


Copyright © 2026, Oracle and/or its affiliates.

```
  <xsd:element name="SGGIMDsEvents">
<xsd:complexType>
      <xsd:all>
 <xsd:element name="D1-DeviceEventSeeder" minOccurs="0">
  <xsd:complexType>
   <xsd:all>
    <xsd:element name="deviceEventId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="14" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="bo" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="boStatus" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="12" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="sender" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalSenderId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="36" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="deviceEventType" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalEventName" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="254" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="eventDateTime" nillable="true" type="xsd:dateTime"
```


Copyright © 2026, Oracle and/or its affiliates.

```
minOccurs="0"/>
    <xsd:element name="eventEndDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0" />
    <xsd:element name="externalTimeZone" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="20" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="deviceId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="12" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="deprecatedDeviceId" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="12" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="creationDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0" />
    <xsd:element name="statusUpdateDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
    <xsd:element name="statusReason" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="rawEventInformation" type="xsd:anyType"
minOccurs="0" />
    <xsd:element name="externalSourceIdentifier" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="120" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="eventInformation" minOccurs="0">
     <xsd:complexType>
      <xsd:all>
       <xsd:element name="externalEventIdentifier" minOccurs="0">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:maxLength value="30" />
         </xsd:restriction>
        </xsd:simpleType>
       </xsd:element>
       <xsd:element name="externalEventCategory" minOccurs="0">
        <xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalEventSeverity" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalDeviceType" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalServiceLocationId" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalCommunicationModuleIdentifier"
minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalGatewayIdentifier" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalStatusValue" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
        </xsd:element>
        <xsd:element name="externalStatusDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
        <xsd:element name="externalCommandId" minOccurs="0">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:maxLength value="30" />
          </xsd:restriction>
         </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xsd:element>
    <xsd:element name="externalEventDescription" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="60" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalEventReason" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="30" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="externalStatusReason" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="60" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="sourceTimeZone" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="10" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="timeZone" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="10" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="dateTimesInStandard" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:enumeration value="D1NO" />
       <xsd:enumeration value="D1YS" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
   </xsd:all>
  </xsd:complexType>
 </xsd:element>
 <xsd:element name="version" minOccurs="0">
  <xsd:simpleType>
   <xsd:restriction base="xsd:decimal">
    <xsd:minExclusive value="-99999" />
    <xsd:maxExclusive value="99999" />
   </xsd:restriction>
  </xsd:simpleType>
 </xsd:element>
 <xsd:element name="deviceIdentifierNumber" minOccurs="0">
```


Copyright © 2026, Oracle and/or its affiliates.

```
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="60" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="newDeviceEvent" minOccurs="0">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:maxLength value="14" />
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
    <xsd:element name="nextRetryDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0" />
    <xsd:element name="retryUntilDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
    <xsd:element name="processData" minOccurs="0">
     <xsd:complexType>
      <xsd:all>
       <xsd:element name="errorEncountered" minOccurs="0">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:enumeration value="D1NO" />
          <xsd:enumeration value="D1YS" />
         </xsd:restriction>
        </xsd:simpleType>
       </xsd:element>
       <xsd:element name="dateTimesInStandard" minOccurs="0">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:enumeration value="D1NO" />
          <xsd:enumeration value="D1YS" />
         </xsd:restriction>
        </xsd:simpleType>
       </xsd:element>
       <xsd:element name="logs" minOccurs="0">
        <xsd:complexType>
         <xsd:all>
          <xsd:element name="logsList" minOccurs="0" >
           <xsd:complexType>
            <xsd:all>
             <xsd:element name="logsEntry" minOccurs="0">
              <xsd:complexType>
               <xsd:all>
                <xsd:element name="sequence" minOccurs="0">
                 <xsd:simpleType>
                  <xsd:restriction base="xsd:decimal">
                   <xsd:minExclusive value="-999" />
                   <xsd:maxExclusive value="999" />
                  </xsd:restriction>
                 </xsd:simpleType>
                </xsd:element>
                <xsd:element name="mo" minOccurs="0">
                 <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <xsd:maxLength value="12" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue1" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue2" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue3" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue4" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="pkValue5" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="logEntryType" minOccurs="0">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="D1TD" />
                    <xsd:enumeration value="F1CR" />
                    <xsd:enumeration value="F1ER" />
                    <xsd:enumeration value="F1EX" />
                    <xsd:enumeration value="F1ST" />
                    <xsd:enumeration value="F1SY" />
                    <xsd:enumeration value="F1TD" />
                    <xsd:enumeration value="F1US" />
                   </xsd:restriction>
                  </xsd:simpleType>
                 </xsd:element>
                 <xsd:element name="logDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0" />
                 <xsd:element name="boStatus" minOccurs="0">
```


Copyright © 2026, Oracle and/or its affiliates.

```
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="12" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="description" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="60" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="user" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="8" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="logMessage" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="4000" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="characteristicType" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="20" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="characteristicValue" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="16" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="adhocValue" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="254" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue1" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue2" minOccurs="0">
```


Copyright © 2026, Oracle and/or its affiliates.

```
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue3" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue4" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="fkValue5" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="50" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageCategory" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:decimal">
   <xsd:minExclusive value="-99999" />
   <xsd:maxExclusive value="99999" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageNumber" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:decimal">
   <xsd:minExclusive value="-99999" />
   <xsd:maxExclusive value="99999" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageParm1" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="30" />
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="messageParm2" minOccurs="0">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:maxLength value="30" />
  </xsd:restriction>
 </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      </xsd:element>
      <xsd:element name="messageParm3" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm4" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm5" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm6" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm7" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm8" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="messageParm9" minOccurs="0">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:maxLength value="30" />
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
     </xsd:all>
    </xsd:complexType>
   </xsd:element>
  </xsd:all>
   <xsd:attribute name="action" type="xsd:string" use="optional" />
 </xsd:complexType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
          </xsd:element>
         </xsd:all>
        </xsd:complexType>
       </xsd:element>
      </xsd:all>
     </xsd:complexType>
    </xsd:element>
   </xsd:all>
  </xsd:complexType>
 </xsd:element>
  <xsd:element name="D1-InitialLoadIMD" minOccurs="0">
    <xsd:complexType>
      <xsd:all>
        <xsd:element name="initialMeasurementDataId" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="14"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="bo" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="fromDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="toDateTime" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="boStatus" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="12"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="statusReason" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="dataSource" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1RE"/>
              <xsd:enumeration value="D1ST"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="timeZone" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
              <xsd:maxLength value="10"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="creationDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="comments" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="254"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isTraceOn" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isIntervalDateTimePopulated" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="serviceProvider" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isAutomatedRetry" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="nextRetryDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="retryUntilDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="veeGroupRole" nillable="true" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1ES"/>
              <xsd:enumeration value="D1IL"/>
              <xsd:enumeration value="D1IS"/>
              <xsd:enumeration value="D1MO"/>
              <xsd:enumeration value="D1PR"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="startReadingDateTime" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="startReading" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:decimal">
              <xsd:minExclusive value="-9999999999.999999"/>
              <xsd:maxExclusive value="9999999999.999999"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="startReadingCondition" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="preVEE" minOccurs="0">
          <xsd:complexType>
            <xsd:all>
              <xsd:element name="simdId" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="14"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="dvcIdN" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="120"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="mcId" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="12"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="mcIdN" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="120"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalId" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="120"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="uom" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalUOM" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="tou" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalTOU" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="sqi" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="externalSQI" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="30"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="stDt" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
              <xsd:element name="stQty" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:decimal">
                    <xsd:minExclusive value="-9999999999.999999"/>
                    <xsd:maxExclusive value="9999999999.999999"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="enDt" nillable="true" type="xsd:dateTime"
minOccurs="0"/>
              <xsd:element name="enQty" minOccurs="0">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:decimal">
                    <xsd:minExclusive value="-9999999999.999999"/>
                    <xsd:maxExclusive value="9999999999.999999"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="imdType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="D1ES"/>
      <xsd:enumeration value="D1GA"/>
      <xsd:enumeration value="D1IL"/>
      <xsd:enumeration value="D1MO"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="mcIS" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="D1IN"/>
      <xsd:enumeration value="D1SC"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="inShift" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="D1NS"/>
      <xsd:enumeration value="D1SH"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="mcm" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:minExclusive value="-999999999999.999999"/>
      <xsd:maxExclusive value="999999999999.999999"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="nd" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:minExclusive value="-999999999"/>
      <xsd:maxExclusive value="999999999"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="tz" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="externalTimeZone" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="20"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="spi" nillable="true" type="xsd:int"
minOccurs="0"/>
              <xsd:element name="ccond" type="xsd:string" minOccurs="0"/>
              <xsd:element name="sts" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="stsL" minOccurs="0"
maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="s" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:decimal">
                                <xsd:minExclusive value="-99999"/>
                                <xsd:maxExclusive value="99999"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                          <xsd:element name="st" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                <xsd:maxLength value="6"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:element name="msrs" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="mL" minOccurs="0"
maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="s" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:decimal">
                                <xsd:minExclusive value="-99999"/>
                                <xsd:maxExclusive value="99999"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                          <xsd:element name="dt" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
                          <xsd:element name="q" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:decimal">
                                <xsd:minExclusive value="-9999999999.999999"/>
                                <xsd:maxExclusive value="9999999999.999999"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                </xsd:restriction>
                              </xsd:simpleType>
                            </xsd:element>
                            <xsd:element name="ue" minOccurs="0">
                              <xsd:simpleType>
                                <xsd:restriction base="xsd:string">
                                  <xsd:enumeration value="D1UE"/>
                                </xsd:restriction>
                              </xsd:simpleType>
                            </xsd:element>
                            <xsd:element name="fc" type="xsd:string"
minOccurs="0"/>
                            <xsd:element name="r" minOccurs="0">
                              <xsd:simpleType>
                                <xsd:restriction base="xsd:decimal">
                                  <xsd:minExclusive value="-9999999999.999999"/>
                                  <xsd:maxExclusive value="9999999999.999999"/>
                                </xsd:restriction>
                              </xsd:simpleType>
                            </xsd:element>
                            <xsd:element name="rc" type="xsd:string"
minOccurs="0"/>
                            <xsd:element name="sts" minOccurs="0">
                              <xsd:complexType>
                                <xsd:sequence>
                                  <xsd:element name="stsL" minOccurs="0"
maxOccurs="unbounded">
                                    <xsd:complexType>
                                      <xsd:sequence>
                                        <xsd:element name="s" minOccurs="0">
                                          <xsd:simpleType>
                                            <xsd:restriction base="xsd:decimal">
                                              <xsd:minExclusive value="-99999"/>
                                              <xsd:maxExclusive value="99999"/>
                                            </xsd:restriction>
                                          </xsd:simpleType>
                                        </xsd:element>
                                        <xsd:element name="st" minOccurs="0">
                                          <xsd:simpleType>
                                            <xsd:restriction base="xsd:string">
                                              <xsd:maxLength value="6"/>
                                            </xsd:restriction>
                                          </xsd:simpleType>
                                        </xsd:element>
                                      </xsd:sequence>
                                    </xsd:complexType>
                                  </xsd:element>
                                </xsd:sequence>
                              </xsd:complexType>
                            </xsd:element>
                          </xsd:sequence>
                        </xsd:complexType>
                      </xsd:element>
                    </xsd:sequence>
                  </xsd:complexType>
                </xsd:element>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xsd:all>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rawData" type="xsd:anyType" minOccurs="0"/>
<xsd:element name="processData" minOccurs="0">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="isShiftedStartEnd" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="isShiftedIntervals" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="isTimeZoneConverted" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="isErrorEncountered" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D1NO"/>
            <xsd:enumeration value="D1YS"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="servicePointId" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="12"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="installationConstant" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:decimal">
            <xsd:minExclusive value="-999999.999999"/>
            <xsd:maxExclusive value="999999.999999"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="deviceId" minOccurs="0">
        <xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <xsd:restriction base="xsd:string">
                      <xsd:maxLength value="12"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="allowNonZeroIntervalsForEstimate"
minOccurs="0">
                  <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="D1NO"/>
                      <xsd:enumeration value="D1YS"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="disableReEstimate" minOccurs="0">
                  <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="D1NO"/>
                      <xsd:enumeration value="D1YS"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="reEstimationActivity" minOccurs="0">
                  <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                      <xsd:maxLength value="14"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:element>
                <xsd:element name="logs" minOccurs="0">
                  <xsd:complexType>
                    <xsd:all>
                      <xsd:element name="logsList" minOccurs="0">
                        <xsd:complexType>
                          <xsd:sequence>
                            <xsd:element name="logsEntry" minOccurs="0">
                              <xsd:complexType>
                                <xsd:sequence>
                                  <xsd:element name="sequence" minOccurs="0">
                                    <xsd:simpleType>
                                      <xsd:restriction base="xsd:decimal">
                                        <xsd:minExclusive value="-999"/>
                                        <xsd:maxExclusive value="999"/>
                                      </xsd:restriction>
                                    </xsd:simpleType>
                                  </xsd:element>
                                  <xsd:element name="mo" minOccurs="0">
                                    <xsd:simpleType>
                                      <xsd:restriction base="xsd:string">
                                        <xsd:maxLength value="12"/>
                                      </xsd:restriction>
                                    </xsd:simpleType>
                                  </xsd:element>
                                  <xsd:element name="pkValue1" minOccurs="0">
                                    <xsd:simpleType>
                                      <xsd:restriction base="xsd:string">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue2" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue3" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue4" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="pkValue5" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="50"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="logEntryType"
minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:enumeration value="D1TD"/>
                                    <xsd:enumeration value="F1CR"/>
                                    <xsd:enumeration value="F1ER"/>
                                    <xsd:enumeration value="F1EX"/>
                                    <xsd:enumeration value="F1ST"/>
                                    <xsd:enumeration value="F1SY"/>
                                    <xsd:enumeration value="F1TD"/>
                                    <xsd:enumeration value="F1US"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
                              <xsd:element name="logDateTime"
nillable="true" type="xsd:dateTime" minOccurs="0"/>
                              <xsd:element name="boStatus" minOccurs="0">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:maxLength value="12"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:element>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                <xsd:element name="description" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="60"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="user" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="8"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="logMessage" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="4000"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="characteristicType"
            minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="20"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="characteristicValue"
            minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="16"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="adhocValue" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="254"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="fkValue1" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="50"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="fkValue2" minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="50"/>
                                    </xsd:restriction>
```


Copyright © 2026, Oracle and/or its affiliates.

```
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="fkValue3" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="50"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="fkValue4" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="50"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="fkValue5" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="50"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="messageCategory"
minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal">
          <xsd:minExclusive value="-99999"/>
          <xsd:maxExclusive value="99999"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="messageNumber"
minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal">
          <xsd:minExclusive value="-99999"/>
          <xsd:maxExclusive value="99999"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="messageParm1"
minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="30"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="messageParm2"
minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="30"/>
        </xsd:restriction>
      </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                </xsd:element>
                                <xsd:element name="messageParm3"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm4"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm5"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm6"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm7"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm8"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="messageParm9"
minOccurs="0">
                                  <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                      <xsd:maxLength value="30"/>
                                    </xsd:restriction>
                                  </xsd:simpleType>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                                </xsd:element>
                              </xsd:sequence>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:all>
                </xsd:complexType>
              </xsd:element>
              <xsd:element name="deviceEventTypes" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="deviceEventTypesList" minOccurs="0"
maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="deviceEventType" minOccurs="0">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                <xsd:maxLength value="30"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:all>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="boStatusDateTime" nillable="true"
type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="isReprocessPerformed" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="D1NO"/>
              <xsd:enumeration value="D1YS"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="serviceProviderExternalId" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="60"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="syncIMDOtherInfo" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="meterReadSource" type="xsd:string"
minOccurs="0"/>
              <xsd:element name="reviewHiLo" type="xsd:string" minOccurs="0"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="version" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:decimal">
              <xsd:minExclusive value="-99999"/>
              <xsd:maxExclusive value="99999"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>

 </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

# Smart Grid Gateway Adapter Native Implementations

This section describes Smart Grid Gateway native implementations. Native implementations can be used with **either** on-premises implementations of Oracle Utilities Smart Grid Gateway, or with Oracle Utilities cloud services such as Oracle Utilities Meter Solution Cloud Service or Oracle Utilities Customer Cloud Service.

- Supported Head End Systems
- Payload Processing
- Smart Meter Commands
- Master Configurations
- Native Implementation Configuration Checklists

## Supported Head End Systems

Oracle Utilities Meter Solution products support integration with a number of vendor-specific head end systems via Oracle Smart Grid Gateway Adapters. The table below lists the supported versions of head end system for each Adapter available with Oracle Utilities Meter Solution products.

| Adapter | Protocol | Head End System Version |
|---|---|---|
| Adapter for Itron OpenWay | Proprietary | Itron OpenWay 3.70, 3.90, 4.1, 6.10<br>OpenWay Collection Manager 5.4 |
| Adapter for Itron Field Collection System | Proprietary | Field Collection System 4.63 |


Copyright © 2026, Oracle and/or its affiliates.

| Adapter | Protocol | Head End System Version |
|---|---|---|
| Adapter for Landis+Gyr | MultiSpeak 3.0<br>CIM 1.0v2<br>CIM 2.0 | Gridstream Command Center 7.5, 8.1, 8.4, 8.5, 9.0.4 |
| Adapter for Sensus RNI | MultiSpeak 4.1 | Sensus RNI 4.4, 4.5.1, 4.10, 4.12, 4.14, 4.15, 4.16 |
| Adapter for Silver Spring Networks (Itron AMM) | Proprietary<br>MultiSpeak 4.1 | UtilityIQ Version 4.4, 4.7, 4.10, 4.14, 5.0, 5.1, 5.2, 6.1 |
| MV-90 Adapter for Itron | Binary file format | N/A<br>SGG supports the .mv9 binary, mainframe data format. |

# Payload Processing

In native implementations, Oracle Smart Grid Gateway adapters process usage and device event payloads via batch processing. Files are sent from a head end system to a file system (for on-premises implementations) or an Oracle Cloud Object Storage location (for cloud services), where they are processed via batch processes.

> **Note**
>
> The total length of the file name for payload files, including the name, delimiter, and extension, should be no greater than 96 characters.

This process requires set up and configuration of the following:

- **Key Rings** and **Key Pairs**: Records used to link an Oracle Cloud Object Storage location to records in Oracle Utilities cloud services. Used with cloud services only.
- **Object Storage Locations**: One or more Oracle Object Storage buckets where payload files are sent from the head end system for processing. Used with cloud services only.
- **File Storage Configuration Extendable Lookup Values**: An extendable lookup value that defines the file location used for payload processing and allows it to be referenced by batch processes
- **Head End Systems**: A record that defines how specific types of payload processing are handled, such the business object to use to create initial measurements and device events
- **Adapter Extendable Lookups**: Lookup value that define details used by data mapping and processing
- **SGG Payload Processing Configuration Extendable Lookup Values**: An extendable lookup value that defines payload processing details for a specific head end system
- **Payload Processing Batch Controls**: A batch control used for payload process (based on a template batch control provided with the base package)

See the following sections below for more details about each of these.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The Itron OpenWay Adapter uses an alternative method for processing usage and device event payloads. See Itron OpenWay Payload Processing for more information.

- Creating Key Rings and Pairs
- Creating Object Storage Locations
- Creating File Storage Extendable Lookup Values
- Creating Head End Systems
- Configuring Adapter Extendable Lookups
- Creating SGG Payload Processing Extendable Lookup Values
- Creating Payload Processing Batch Controls
- Mapping Additional Fields
- Itron OpenWay Payload Processing
- Itron Field Collection System Payload Processing
- Adapter Development Kit Custom Payload Processing
- Adapter Development Kit Common Information Model (CIM) Payload Processing

## Creating Key Rings and Pairs

> **Note**
>
> Key Rings and Pairs are used with cloud services **only**.

When using Smart Grid Gateway adapters with cloud services, connection between the Oracle Utilities cloud service and Oracle Object Storage requires an API signature key. See **API Key Management** in the *Oracle Utilities Cloud Services Object Storage Setup Guide* for more information.

API key rings and key pairs are maintained in the **Key Ring** portal. This portal contains the following zones:

- **Key Ring**: Displays basic information about the key ring
- **Key Pairs**: Displays a list of key pairs for the current key ring

Key rings are defined by the following:

- **Key Ring**: A unique code for the key ring
- **Key Ring Class**: Signature (default)
- **Status**: The current status of the key ring. Note: Key pairs can only be generated for Active key rings. Once a key ring has been deactivated, you can no longer create key pairs for that ring.
- **Description**: A name for the key ring (this will be referenced in the File Location extendable lookup)


Copyright © 2026, Oracle and/or its affiliates.

Once the key ring is created, you need to generate and activate the key pair. Click **Generate Key** to generate a key pair for the key ring.

Key Pairs are defined by the following:

- **Sequence**: The sequence of the key pair (the order in which the key pair was created)
- **Creation Date/Time**: The date/time when the key pair was created
- **Key Status**: The current status of the key pair. Key pairs are inactive when first created.
- **Public Key**: Click **View** to open a dialog box containing the public key.
- **Action**: Click **Activate** to activate an inactive key pair.

Once the key ring and key pair have been created, the key should be pasted into the **User API Key** in **Oracle Identification and Access Management** (IAM) See the **User API Keys** section in the **Security and Access Management** section of the **Managing Object Storage** chapter in the *Oracle Cloud Infrastructure Services* documentation. To copy the public key click **View** in the **Actions** column in the **Key Pairs** zone, and select and copy the text in the **View Public Key** dialog box.

The last step is to activate the key. Click **Activate** in the **Actions** column in the **Key Pairs** zone. A dialog box opens displaying the following message: “Warning(s): Activating a key assumes that you have already registered the public key with the appropriate third parties. Press Cancel to abort.” Note: Be sure to copy the public key to **Oracle Identification and Access Management** before activating it. Click **OK** to activate the key. The **Key Status** column will change to “Active”.

## Creating Object Storage Locations

> **Note**
>
> Object Storage Locations are used with cloud services **only**.

When using Smart Grid Gateway adapters with cloud services, you need to create one or more buckets in Oracle Object Storage where payload files will be sent for processing. Refer to the *Oracle Utilities Cloud Services Object Storage Setup Guide* and the *Oracle Cloud Infrastructure* documentation for more information about creating buckets in Oracle Object Storage.

After creating the buckets in Oracle Object Storage, make a note of the following, as this information is needed in later configuration steps:

- Namespace
- User Oracle Cloud ID (User OCID)
- Tenancy Oracle Cloud ID (Tenancy OCID)
- Compartment Oracle Cloud ID (Compartment OCID)

### Suggested Buckets for Payload Processing

This section outlined suggested sets of Object Storage buckets for each of the Smart Grid Gateway adapters.

### General Guidelines


Copyright © 2026, Oracle and/or its affiliates.

As a general rule, each adapter should have separate buckets for each of the following:

- Usage payloads
- Usage archive
- Usage errors
- Device Event payloads
- Device Event archive
- Device Event errors

If an adapter uses payloads in more than one format (such as in CSV and XML formats), you should also consider creating separate buckets for each file format type.

> **Note**
>
> If possible, archive buckets should be created in an “Archive” storage tier.

### Adapter Development Kit

The table below lists suggested Object Storage buckets for the Adapter Development Kit:

| Bucket Name | Storage Tier | Smart Grid Gateway Description |
|---|---|---|
| adk-csv | Standard | SGG ADK Adapter - CSV input |
| adk-csv-arch | Archive | SGG ADK Adapter - CSV archive |
| adk-csv-error | Standard | SGG ADK Adapter - CSV errors |
| adk-seeder | Standard | SGG ADK Adapter - IMD Seeder input |
| adk-seeder-arch | Archive | SGG ADK Adapter - IMD Seeder archive |
| adk-seeder-error | Standard | SGG ADK Adapter - IMD Seeder errors |
| adk-xml | Standard | SGG ADK Adapter - XML input |
| adk-xml-arch | Archive | SGG ADK Adapter - XML archive |
| adk-xml-error | Standard | SGG ADK Adapter - XML errors |

### Adapter for Itron OpenWay

The table below lists suggested Object Storage buckets for the Adapter for Itron OpenWay:

| Bucket Name | Storage Tier | Smart Grid Gateway Description |
|---|---|---|
| itronexception | Standard | SGG Itron OW Adapter - event input |
| itronexception-arch | Archive | SGG Itron OW Adapter - event archive |
| itronexception-error | Standard | SGG Itron OW Adapter - event errors |
| itronxml | Standard | SGG Itron OW Adapter - XML input |
| itronxml-arch | Archive | SGG Itron OW Adapter - XML archive |
| itronxml-error | Standard | SGG Itron OW Adapter - XML errors |

### MV90 Adapter for Itron

The table below lists suggested Object Storage buckets for the MV90 Adapter for Itron


Copyright © 2026, Oracle and/or its affiliates.

| Bucket Name | Storage Tier | Smart Grid Gateway Description |
|---|---|---|
| mv90-usage | Standard | SGG MV90 Adapter - usage input |
| mv90-usage-arch | Archive | SGG MV90 Adapter - usage archive |
| mv90-usage-error | Standard | SGG MV90 Adapter - usage errors |

### Adapter for Landis+Gyr

The table below lists suggested Object Storage buckets for the Adapter for Landis+Gyr:

| Bucket Name | Storage Tier | Smart Grid Gateway Description |
|---|---|---|
| lg-cim-event | Standard | SGG Landis+Gyr Adapter - CIM input |
| lg-cim-event-arch | Archive | SGG Landis+Gyr Adapter - CIM archive |
| lg-cim-event-error | Standard | SGG Landis+Gyr Adapter - CIM errors |
| lg-event | Standard | SGG Landis+Gyr Adapter - event input |
| lg-event-arch | Archive | SGG Landis+Gyr Adapter - events archive |
| lg-event-error | Standard | SGG Landis+Gyr Adapter - event errors |
| lg-usage | Standard | SGG Landis+Gyr Adapter - usage input |
| lg-usage-arch | Archive | SGG Landis+Gyr Adapter - usage archive |
| lg-usage-error | Standard | SGG Landis+Gyr Adapter - usage errors |

### Adapter for Sensus

The table below lists suggested Object Storage buckets for the Adapter for Sensus:

| Bucket Name | Storage Tier | Smart Grid Gateway Description |
|---|---|---|
| sensus-event | Standard | SGG Sensus Adapter - event input |
| sensus-event-arch | Archive | SGG Sensus Adapter - events archive |
| sensus-event-error | Standard | SGG Sensus Adapter - event errors |
| sensus-usage | Standard | SGG Sensus Adapter - usage input |
| sensus-usage-arch | Archive | SGG Sensus Adapter - usage archive |
| sensus-usage-error | Standard | SGG Sensus Adapter - usage errors |

### Adapter for Silver Spring Networks

The table below lists suggested Object Storage buckets for the Adapter for Silver Spring Networks:

| Bucket Name | Storage Tier | Smart Grid Gateway Description |
|---|---|---|
| ssn-csv | Standard | SGG SSN Adapter - CSV input |
| ssn-csv-arch | Archive | SGG SSN Adapter - CSV archive |
| ssn-csv-error | Standard | SGG SSN Adapter - CSV errors |
| ssn-ssnxml | Standard | SGG SSN Adapter - XML input |
| ssn-ssnxml-arch | Archive | SGG SSN Adapter - XML archive |
| ssn-ssnxml-error | Standard | SGG SSN Adapter - XML errors |


Copyright © 2026, Oracle and/or its affiliates.

## Creating File Storage Extendable Lookup Values

Payload processing requires defining details of file storage locations. These are file system or Object Storage locations where payload files are posted for processing. File storage configuration information is captured as values for the File Storage Configuration (F1-FileStorage) extendable lookup. These records will be referenced by the batch controls that will process the payloads.

Values for this extendable lookup are defined by the following:

- **Value**: A unique code for the extendable lookup value. This value will be referenced as a batch control parameter value.
- **Description**: A description of the extendable lookup value
- **Status**: The current status of the value. Select “Active”.
- **File Storage Details**: This section defines details for the file storage location, including:
  - **File Adapter**: The type of file adapter for the location. Options include:
    * Native File Storage: Used for on-premises implementations. Additional details for this type of adapter include:
      * **File Path**: A valid file path to the file location on the file system where payload files will be found
    * Oracle Cloud Object Storage: Used with Oracle Utilities cloud services. Additional details for this type of adapter include:
      * **User**: The user Oracle Cloud ID (ODIC) for the object storage location
      * **Tenancy**: The tenancy Oracle Cloud ID (ODIC) for the object storage location
      * **Compartment**: The compartment Oracle Cloud ID (ODIC) for the object storage location
      * **Namespace**: The namespace for the object storage location
      * **Key Ring**: The Key Ring you created earlier
      * **Region**: The region of the object storage tenancy for the connection (Values for this field are defined in the F1_REGION_FLG lookup.
      * **Bucket Name Prefix**: Prefix used by Object Storage Bucket configuration

Refer to **External File Storage** in the *Oracle Utilities Application Framework Administrative User Guide* and the *Oracle Utilities Cloud Services Object Storage Setup Guide* for more information about using Oracle Object Storage with cloud service implementations.

## Creating Head End Systems

The next step is to create a record for the head end system. This record defines processing methods for the head end system.

**Processing Methods for Payload Processing**

The following processing methods should be configured to support payload processing:

- **Device Event Mapping**: defines how head end-specific device events are mapped to standard device event names. The Device Event Seeder business object uses this


Copyright © 2026, Oracle and/or its affiliates.

processing method to determine which type of device event business object to instantiate when receiving device events from a head-end system.

- **Initial Measurement Creation**: defines the business object(s) and other data used to create initial measurements. When using Legacy Measurement Processing, the IMD Seeder business object uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from a head end system. When using Direct Measurement Processing, specify an IMD Type(s). See Direct Measurement Processing in the *Business User Guide* for more information.
- **UOM Translation**: defines how head end-specific unit of measure (UOM) codes are mapped to standard UOM codes.

If usage payload include time of use periods or service quantity identifiers, you should also configure **SQI Translation** and **TOU Translation** processing methods.

See Head End Systems and Understanding Processing Methods in the *Administrative User Guide* for more information about creating and configuring a head end system and its processing methods.

## Configuring Adapter Extendable Lookups

Smart Grid Gateway adapters use extendable lookups to define data mapping and processing rules specific to each adapter. These include:

- **Device Event Mapping**: used to map head end specific device event codes to standard event names
- **UOM Code to Standard UOM Mapping**: used to map head end specific units of measure (UOM) to standard UOM codes
- **Status Code Mapping**: used to map head end specific interval status codes to standard Condition codes.

Some adapters have additional extendable lookups used to define categories, statuses, and other data. See the following topics for information about adapter-specific extendable lookups:

- Configuring Adapter Development Kit Extendable Lookups
- Configuring Itron OpenWay Extendable Lookups
- Configuring Landis+Gyr Extendable Lookups
- Configuring MV90 Extendable Lookups
- Configuring Sensus Extendable Lookups
- Configuring Silver Spring Networks Extendable Lookups
- Configuring Itron Field Collection System Extendable Lookups
- Configuring Itron FCS Extendable Lookups

## Configuring Itron FCS Extendable Lookups

This section outlines some of the extendable lookups that must be configured for use with the ItronField Collection System adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Note: In the descriptions below, references to "the system" are references to Oracle Utilities Meter Solution applications, including Meter Data Management, Smart Grid Gateway.


Copyright © 2026, Oracle and/or its affiliates.

### Meter Read Download Extendable Lookups

The following extendable lookups are used when creating meter read download files to be sent to FCS.

#### FCS Customer Special Code Mapping

The FCS - Customer Special Code Mapping (D1-CustomerSpecialCodeMapLkup) extendable lookup is used to map special customer codes used in Itron FCS to specific flag settings. These flag settings are used to provide warnings of potential hazards to employees in the field. Note that a single value for this extendable lookup can contain multiple mappings. Values for this extendable lookup include the following:

- **Customer Special Code Mapping**: Mappings between special codes and related flags.
  - **Special Code**: The special code for the mapping.
  - **Priority**: The priority for the mapping.
  - **Beep Flag**: A flag that indicates (True or False) if an audible "beep" warning should be issued for the code.
  - **Vibrate Flag**: A flag that indicates (True or False) if a vibration warning should be issued for the code.
  - **Flash Screen Flag**: A flag that indicates (True or False) if visible flashing warning should be issued for the code.

#### FCS Read Type

The FCS - Read Type (D1-FCSReadTypeLookup) extendable lookup is used to map specific details to read types defined for FCS meters. Note that a single value for this extendable lookup can contain multiple mappings. Values for this extendable lookup include the following:

- **Read Type Mapping - FCS Mapping Information**: Mappings between read types and meter configuration details.
  - **Read Type Code**: The read type for the mapping
  - **Text Prompt**: The value of the textPrompt value
  - **Read Type Description**: A description of the read type (such as "KWH")
  - **Read Methods**: The method used to perform the read on this meter (such as "Manual" or "Mobile Meter Read")
  - **Number Dials**: The number of dials on the meter
  - **Number of Digits Right**: The number of decimal values
  - **CIS Division**: The CIS Division associated with the read type

#### FCS Meter Type Register Group Mapping

The FCS - Meter Type Register Group Mapping (D1-MeterTypeRegGrpMapLookup) extendable lookup is used to map device configuration types to specific details concerning the meter type. Note that a single value for this extendable lookup can contain multiple mappings. Values for this extendable lookup include the following:

- **Meter Type Mapping**: Mappings between device configuration types and meter type details
  - **Device Configuration Type**: The device configuration for the mapping
  - **Read Methods**: The method used to perform the read on this meter (such as "Manual" or "Mobile Meter Read")


Copyright © 2026, Oracle and/or its affiliates.

- **Number Dials**: The number of dials on the meter
- **Number of Decimals**: The number of decimal values captured on meter readings
- **Meter Type**: An FCS code for the meter type
- **Register Group**: An FCS code for the register group

**FCS Rate to Hi-Low % Mapping**

The FCS - Rate To Hi-Low % Mapping (D1-RateToHighLowMappingLookup) extendable lookup to map FCS rate codes to maximum and minimum usage levels (labeled H1, H2, L1, and L2). Note that a single value for this extendable lookup can contain multiple mappings. Values for this extendable lookup include the following:

- **Rate High to Low % Mapping**: Mappings between rate codes and high/low usage levels
  - **Rate Code**: The FCS rate code for the mapping
  - **H1%**: The percentage that represents the "H1" usage level
  - **H2%**: The percentage that represents the "H2" usage level
  - **L1%**: The percentage that represents the "L1" usage level
  - **L2%**: The percentage that represents the "L2" usage level

**FCS TIM Name Mapping**

The FCS - TIM Name Mapping (D1-TIMNameMappingLookup ) extendable lookup is used to map Manufacturer and Model pairings to Itron FCS Translation Interface Module (TIM) names. TIMs are software components within Itron's systems (like MV-90) that enable communication with and data collection from a wide range of meter types, in particular with time of use and interval meters. Note that a single value for this extendable lookup can contain multiple mappings. Values for this extendable lookup include the following:

- **TIM Name Mapping**: Mappings between Manufacturer/Model pairs and TIM names and password.
  - **Manufacturer**: A manufacturer defined in the system
  - **Register Model**: A model, based on the selected Manufacturer
  - **TIM Name**: The TIM name associated with the Manufacturer and Model
  - **Password**: A password used to communicate with the TIM

### Payload Processing Extendable Lookups

The following extendable lookups are used when processing measurement payloads received from FCS.

**FCS UOM Code Standard UOM Code Mapping**

Tee FCS - UOM Code Standard UOM Code Mapping (D9-FCSHeadendUOMLookup) extendable lookup...

Usage received from the Itron FCS system uses specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The FCS UOM Code Standard UOM Mapping (D1-FCSHeadendUOMLookup) extendable lookup is used for this purpose. Each value defined for this extendable lookup should include the following:

- **Head-End Unit of Measure**: The unit of measure code used by the Itron FCS application
- **Description**: A description of the unit of measure code.
- **Status**: The status of the lookup value (can be Active or Inactive)


Copyright © 2026, Oracle and/or its affiliates.

- **Unit of Measure**: The unit of measure defined in the system.
- **Time of Use**: An optional time of use period, defined in the system, used to further distinguish the unit of measure.
- **Service Quantity Identifier**: An optional service quantity identifier, defined in the system, used to further distinguish the unit of measure.

**FCS Head-End Time Zone to Standard Mapping**

The FCS - Head-End Time Zone to Standard Mapping (D9-FCSTimeZoneTransLookup) extendable lookup maps time zones defined in the Itron FCS application to time zones defined in Oracle Utilities Smart Grid Gateway and Meter Data Management. Each value defined for this extendable lookup should include the following:

- **Head-End Time Zone**: The time zone used by the Itron FCS application
- **Description**: A description of the time zone
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Time Zone**: The time zone defined in the system

**FCS Upload Parameters**

The FCS - Upload Parameters (D9-FCSUploadParameters) extendable lookup is used to map specific types of FCS readings to read conditions, and specific types of FCS failures (based on read conditions in the FCS payload) to device event types defined in the FCS Device Event Mapping extendable lookup. For failure types, if a read contains the failure condition, the system creates a reader remark device event based on the specific device event type mapped. This extendable lookup is referenced as the value for the uploadParametersLookup dynamic parameter on the SGG Payload Processing Configuration extendable lookup value used with FCS (see Adapter for Itron Field Collection System (FCS) in Payload Handler Classes and Parameters).

Each value defined for this extendable lookup should include the following:

- **Channel Condition Codes**: Mappings of FCS reading types (such as Regular Read, Verified Read, Bad Read, and so on) to measurement condition codes (from the Measurement Condition extendable lookup).
- **Read Condition Failures**: Mappings of specific FCS failures (such as H1, H2, L1, and L2) to device event types (from the FCS Device Event Mapping extendable lookup)

**FCS Device Event Mapping**

The FCS - Device Event Mapping (D9-DeviceEventMappingLookup) extendable lookup is used to map FCS device event codes into standard event codes to determine which type of device event business object to instantiate when receiving device events from the head-end system. This mapping is done via pre-processing in the Device Event Seeder business object. Each value defined for this extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the head-end system
- **Description**: A description of the device event
- **Status**: The status of the lookup value (can be Active or Inactive)
- **Standard Event Name**: The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

## Creating SGG Payload Processing Extendable Lookup Values


Copyright © 2026, Oracle and/or its affiliates.

You also have to create an extendable lookup value to define how usage and event payloads will be processed.

Payload processing configuration information is captured in the SGG Payload Processing Configuration (D1-SGGPayloadProcess) extendable lookup.

Values for this extendable lookup are defined by the following:

- **Value**: A unique code for the extendable lookup value. This value will be referenced as a batch control parameter value.
- **Description**: A description of the lookup value
- **Status**: The current status of the value. Set this to “Active”.
- **Processing Details**: This section defines details for payload processing, including
  - **Payload Handler Class Name**: The Java class name for the processing handler to be used. See Payload Handler Classes and Parameters below for more information.
  - **Head-End System**: The head end system for which the configuration will apply. Select from the drop-down list.
  - **Populate Raw**: A checkbox that indicates if the payload data is populated as raw data.
  - **Payload Processing Result Type**: The type of payload to which the configuration applies. Options include:
    * Device Events
    * Initial Measurements
    * Initial Measurements and Device Events
  - **Use IMD Type**: A checkbox that indicates if payload processing should use Direct Measurement Processing and IMD Types. If checked, the **IMD Seeder BO (Interval)** and **IMD Seeder BO (Scalar)** fields are hidden. See Direct Measurement Processing in the *Business User Guide* for more information. This checkbox is enabled by default.
  - **Passthrough Date/Time Format**: The date/time format for passthrough measurements. Valid values are "OUAF" and "XSD". When XSD is selected the data will be exported in UTC. Used when using Measurement Data Filtering with Direct Measurement Processing. See Direct Measurement Processing in the *Business User Guide* for more information.
  - **Use IMD Type**: A checkbox that indicates if payload processing should use Direct Measurement Processing and IMD Types. If checked, the **IMD Seeder BO (Interval)** and **IMD Seeder BO (Scalar)** fields are hidden. See Direct Measurement Processing in the *Business User Guide* for more information..
  - **IMD Seeder BO (Interval)**: The seeder business object used to create interval initial measurements. The base package includes the “IMD Seeder” (D1-IMDSeeder) business object for this. Applicable only if the Payload Processing Result Type is “Initial Measurements” or “Initial Measurements and Device Events”.
  - **IMD Seeder BO (Scalar)**: The seeder business object used to create scalar initial measurements. The base package includes the “IMD Seeder” (D1-IMDSeeder) business object for this. Applicable only if the Payload Processing Result Type is “Initial Measurements” or “Initial Measurements and Device Events”.
  - **Device Event Seeder BO**: The seeder business object used to create device events. The base package includes the “Device Event Seeder” (D1-DeviceEventSeeder) business object for this. Applicable only if the Payload Processing Result Type is “Device Events” or “Initial Measurements and Device Events”.


Copyright © 2026, Oracle and/or its affiliates.

  - **Filter IMD**: A checkbox that indicates if initial measurements should be filtered based on unit of measure. Applicable only if the Payload Processing Result Type is “Initial Measurements” or “Initial Measurements and Device Events”. See Filtering Payloads by Unit of Measure or Device Event Mapping for more information. Note that filtering by unit of measure can be used in parallel with Measuring Component Filtering.
  - **Filter Device Events**: A checkbox that indicates if device events should be filtered. Applicable only if the Payload Processing Result Type is “Device Events” or “Initial Measurements and Device Events”. See **Filtering Payloads** below for more information.

- **Dynamic Parameters**: A list of parameter names and values that support additional processing rules, based on the specified Payload Handler Class. See Payload Handler Classes and Parameters below for specific parameters supported by each adapter.
- **User Exit Interceptors**: Defines custom Groovy Library Scripts to be executed via user exit calls when parsing and transforming incoming usage and device data in custom and non-standard formats. User exit interceptors are defined as follows:

| Payload Processing User Exit Type | Groovy Library Script | Groovy Library Method |
|---|---|---|
| On Get Parser | Parsing script | getParser |
| On Get Transformer | Transforming script | getTransformer |
| On Result List | Handler script | getHandler |

See Adapter Development Kit Custom Payload Processing for more information about the On Get Parser and On Get Transformer User Exit Interceptors. See Mapping Additional Fields for information about the On Result List User Exit Interceptor.

> **Note**
>
> On Get Parser and On Get Transformer User Exit Interceptors are applicable only when the **Payload Handler Class Name** is set to `com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits`.

- Payload Handler Classes and Parameters
- Filtering Payloads by Unit of Measure or Device Event Mapping
- Upload Statistics Activities

## Payload Handler Classes and Parameters

This section provides valid Payload Handler Class Names and associated dynamic parameters for supported head end systems and payload types.

### Adapter Development Kit

As delivered, the Adapter Development Kit supports payload processing of usage and event data exported from an AMI head end system in the “native” initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects). See The Adapter Development Kit Native Format for more information.

**Payload Type**: ADK CSV


Copyright © 2026, Oracle and/or its affiliates.

- **Payload Handler Class**: com.splwg.d1.domain.sgg.dg.processing.CSVPayloadHandler

**Payload Type**: ADK XML

- **Payload Handler Class**: com.splwg.d1.domain.sgg.dg.processing.XMLPayloadHandler

**Payload Type**: ADK Seeder

- **Payload Handler Class**: com.splwg.d1.domain.sgg.dg.processing.SeederPayloadHandler

**Processing Custom and Non-Standard Formats**

The Adapter Development Kit can be configured to support payload processing of data from AMI head end systems in custom and non-standard formats. Processing payloads of this type requires a specific **Payload Handler Class** (see below), and creation of Groovy Library Scripts to parse and transform data into the "native" format. See **Adapter Development Kit Custom Payload Processing** and **Payload Processing User Exit Interceptor Scripts** for more information.

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits

The Adapter Development Kit also supports payload processing of data in the Common Information Model (CIM) format. See Adapter Development Kit Common Information Model (CIM) Payload Processing for more information.

### Adapter for Itron OpenWay

**Payload Type**: Itron OpenWay XML

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.d8.processing.ItronXMLPayloadHandler
- **Dynamic Parameters**:

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| intervalValueDecPlaces | Optional value. Specifies the number of digits after the decimal place for the quantity value. If this parameter is missing, the processing logic defaults the number of digits after the decimal place to 6. |
| filterRegisterSource | Optional value. Determines if source register data should be filtered. Valid values are 'true' and 'false' (default) |

**Payload Type**: Itron OpenWay Exception

- **Payload Handler Class**:
  com.splwg.d1.domain.sgg.d8.processing.ExceptionPayloadHandler

### MV90 Adapter for Itron

**Payload Type**: MV90

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d5.processing.MV90PayloadHandler
- **Dynamic Parameters**:


Copyright © 2026, Oracle and/or its affiliates.

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| usePhysicalChannel | Optional value. Determines whether the physical channel ID is passed to the IMD seeder to create the measuring component identifier number. If this is set to false, the MV90 LOGCHAN field is used. |
| fieldForDvcIdN | Optional value. Specifies which field is used as the value for the device ID. Valid values include:<br>• DC_RECID<br>• DC_CUSTID<br>• DC_METERID (default) |
| MV90ScalarChannelSuffix | Optional value. Holds a suffix value to be added to the measuring component identifier number when a scalarIMD is created for register reads. Default value is "_S" |
| processMV90ScalarData | Optional value. Determines if register reads are processed. Valid values are 'true' and 'false' (default). |
| sendStatusAsIs | Optional value. Determines if the status codes will be passed as they came from the head end. If this set to false or not provided, then the default existing behavior when a status code is parsed and status types passed to the IMD seeder in which bits are turned on. Valid values are 'true' and 'false' (default). |
| MV90DateFormat | Optional value. Holds the date format used to parse provided date-time fields. The code supports the limited list of values for the format. If this parameter is missing or if the value supplied to it does not match the values from this list then the format used will be MDDYYhhmm, the default value. Valid values include:<br>• MMDDYYhhmm (default)<br>• MMYYDDhhmm<br>• DDMMYYhhmm<br>• DDYYMMhhmm<br>• YYMMDDhhmm<br>• YYDDMMhhmm |
| MV90IntervalValueDecPlaces | Optional value. Specifies the number of digits after the decimal place for the quantity value. If this parameter is missing, the processing logic defaults the number of digits after the decimal place to 6. |
| MV90StrtmtrAndStopmtrToDecimal | Optional value. Determines if the MV90 STRTMTR (Start Reading Meter) and STOPMTR (Stop Reading Meter) will be converted to a decimal. The processing logic will treat these fields as decimal use the setting to determine how many decimal points are required. |


Copyright © 2026, Oracle and/or its affiliates.

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| usePulseMultiplierAndOffset | Optional value. If set to "true", specifies the pulse values are multiplied by the pulse mutlipler.<br>Valid values include:<br>• true<br>• false (default)<br><br>**Note**<br>This setting only applies to MV90 files that have a Pulse datatype. |

### Adapter for Itron Field Collection System (FCS)

**Payload Type**: Itron FCS

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d9.processing.FCSPayloadHandler
- **Dynamic Parameters:**

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| uploadParametersLookup | Used to specify the FCS Upload Parameters extendable lookup value use to map specific types of FCS readings to read conditions, and specific types of FCS failures (based on read conditions in the FCS payload) to device event types. |

### Adapter for Landis+Gyr

**Payload Type**: L+G Usage

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d3.processing.LGUsagePayloadHandler
- **Dynamic Parameters:**

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| dateTimeInUTC | Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are ‘true’ and ‘false’ (default) |

**Payload Type**: L+G Event

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d3.processing.LGEventPayloadHandler
- **Dynamic Parameters:**


Copyright © 2026, Oracle and/or its affiliates.

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| dateTimeInUTC | Indicates whether the Landis+Gyr system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are ‘true’ and ‘false’ (default) |

**Payload Type**: L+G CIM Event

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d3.processing.LGCIMEventPayloadHandler
- **Dynamic Parameters:** N/A

### Adapter for Sensus

**Payload Type**: Sensus RNI Usage

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d6.processing.SensusUsagePayloadHandler
- **Dynamic Parameters:**

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| dateTimeInUTC | Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are ‘true’ and ‘false’ (default). |
| useExternalTOU | Applies to Scalar usage only.<br>Indicates whether or not an externally referenced TOU period should be used when processing measurement data.<br>When set to true the TOU period Sensus provides will be passed to the IMD Seeder as an external TOU “preVEE/externalTOU”. When not provided or set to false then the TOU period Sensus provides will be passed to the IMD Seeder as the external MC identifier number “preVEE/mcIdN”. Valid values are ‘true’ and ‘false’ (default). |
| sendStatusAsIs | Determines if the status codes will be passed as they received from the head end. When set to true status codes will be sent as they are received from the head end. When not provided or set to false the status codes from the head will be processed through a bit mask to produce multiple individual statuses. Valid values are ‘true’ and ‘false’ (default). |
| sendIntervalDateTimes | When set to true the “preVEE/msrms/mL/dt” element of the IMD seeder for interval records will be populated with the interval date/time and the element “preVEE/enDt” will be populated with the interval date/time of the last interval. When not provided or set to false the “preVEE/msrms/mL/dt” element of the IMD seeder for interval records will not be populated and the element “preVEE/enDt” will be populated with: start date + spi * number of intervals. Valid values are ‘true’ and ‘false’ (default). |


Copyright © 2026, Oracle and/or its affiliates.

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| scalarExpanding | Optional Value. Defines the number of register reads to process in scalar payloads. When defined, Sensus payload processing uses this setting to ensure that all the scalar reads included in a payload will be processed.<br>Default value is 2. |

**Payload Type**: Sensus RNI Event

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d6.processing.SensusEventPayloadHandler
- **Dynamic Parameters:**

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| dateTimeInUTC | Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device. Valid values are 'true' and 'false' (default) |

### Adapter for Silver Spring Networks

**Payload Type**: Silver Spring Networks CSV

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d7.processing.SSNCSVPayloadHandler

**Payload Type**: Silver Spring Networks XML

- **Payload Handler Class**: com.splwg.d1.domain.sgg.d7.processing.SSNXMLPayloadHandler

### Common Parameters

The following parameters can be used by all of the supported head end system adapters.

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| suppressPayloadStatistics | Optional parameter. Determines if the PayloadStatistics business object will not be created while processing payload. If this set to false or not provided, then the default existing behavior when a PayloadStatistics business object is created at the start of the payload processing. Valid values are 'true' and 'false' (default). |
| suppressPayloadError | Optional parameter. Determines if the PayloadErrorNotification business object will not be created when an error occurs while processing payload. If this set to false or not provided, then the default existing behavior when a PayloadErrorNotification business object is created for every error occurred during the payload processing. If this set to false, the error description will be written into the application log file. Valid values are 'true' and 'false' (default). |


Copyright © 2026, Oracle and/or its affiliates.

| Dynamic Parameter Name | Description and Valid Values |
|---|---|
| suppressPayloadSummary | Optional parameter. Determines if the PayloadSummary business object will not be created while processing payload. If this set to false or not provided, then the default existing behavior when a PayloadSummary business object is created at the end of the payload processing. Valid values are ‘true’ and ‘false’ (default). |
| createFilePerCommit | Optional parameter. Determines how files are created when streaming files out to a file system or Object Storage. When set to true, streamed out files are written per commit. Valid values are ‘true’ and ‘false’ (default). |
| commitSizeInKB | Optional Parameter. Specifies the total amount of memory (in KB) allocated for Device Event and IMD seeders before they are committed. Seeders are kept in memory until their size exceeds the value defined for this parameter or the end of payload is reached.<br>Default value is 100.<br>Note: This parameter (either the default value or a specified value) is ignored when the **chunkSize** parameter is set. |
| chunkSize | Optional parameter. Specifies the size in bytes of a portion (chunk) of the payload that will be processed in a single batch “thread”.<br>If parameter is set, the **commitSizeInKB** parameter is ignored.<br>This parameter is supported by the following payload formats:<br>• Landis+Gyr Usage<br>• Landis+Gyr Events<br>• MV90 Usage<br>• Sensus RNI Usage<br>• Sensus RNI Events<br>• Silver Spring Networks Usage (CSV)<br>• Adapter Development Kit Usage sample format (CSV)<br>• Adapter Development Kit via User Exits (if an instance returned from the getParser method implements the `com.splwg.d1.domain.sgg.dg.processing.ChunkProcessor` interface). |

## Filtering Payloads by Unit of Measure or Device Event Mapping

When **Filter IMD** is selected, only measurements whose <externalUOM> matches one of the values defined in the head end system’s UOM Code to Standard UOM Mapping extendable lookup are passed into the system for processing.

When **Filter Device Events** is selected, only device events whose <externalEventName> matches one of the values defined in the head end system’s Device Event Mapping extendable lookup are passed into the system for processing.

## Upload Statistics Activities


Copyright © 2026, Oracle and/or its affiliates.

Upload statistics activities are activities that contain statistics related to the processing of usage and device events sent from a head end system.

There are three types of upload statistics activities:

- **Payload Statistics**: Contains statistics related to a specific payload (file) containing one or more initial measurements or device events.
- **Payload Error Notification**: Contains details concerning processing errors encountered in an individual payload (file) containing one or more initial measurements or device events.
- **Payload Summary**: Contains processing summary statistics for an individual payload (file) containing one or more initial measurements or device events. Payload Summary used to update related payload statistics upon the completion of payload processing.

Upload statistics activities are created during processing of payload files as follows:

- When processing begins for a payload, a **Payload Statistics** activity is created to record the process.
- •If an error occurs during processing, a **Payload Error Notification**activity is created.
- •When payload processing is complete, a **Payload Summary**activity is created, which in turn, updates the Payload Statistics activity with details concerning the processing of the payload, including (the start and end time of the processing, the total processing time, the number of initial measurements or device events processed, and the number of initial measurement or device event errors (if any).

#### Suppressing Upload Statistics Activities

Creation of upload statistics activities can be suppressed through the use of the following dynamic parameters in the SGG Payload Processing Configuration Extendable Lookup:

- suppressPayloadStatistics
- suppressPayloadSummary
- suppressPayloadError (**Note**: If this set to false, the error description will be written into the application log file.)

See **Common Parameters** in Payload Handler Classes and Parametersfor more information about these parameters.

## Creating Payload Processing Batch Controls

The last step is to create payload processing monitor batch controls based on templates. You can create separate batch controls for each head end system or for specific payload types, or any combination of both. For example, you might create a separate batch control for Itron OpenWay usage payloads and device event payloads. Creating specific batch controls allows more control when setting up batch process scheduling for payload processing. For instance, you may want to run usage payload processing every hour, but device payload processing every two hours.

The base package includes the following Payload Processing template batch controls:

- Payload Processing Monitor Template (D1-PLPRM): Used with both Legacy Measurement Processing and Direct Measurement Processing
- Payload Processing Monitor Template with Export (D1-PLPSO): Used with Direct Measurement Processing and required when using Measuring Component Filtering. When used with filtering, this batch process can also send initial measurement payloads to a file system or an object storage location.


Copyright © 2026, Oracle and/or its affiliates.

- SGG ML Payload Processing Monitor Template (D1-PLMLP): Used with Machine Learning Anomaly Scoring to retrieve measurements following machine learning processing (see Configuring Machine Learning Anomaly Scoring for more information).

To create a payload processing monitor batch control, search for the appropriate template batch control, click **Duplicate**, enter a unique code for the duplicate batch control and click **OK**.

This new batch control can be used to process payloads received from head end systems.

### Payload Processing Batch Control Parameters

Payload processing batch processing is controlled by the following parameters:

| Parameter Name | Description | Detailed Description | Required? |
|---|---|---|---|
| includeFiles | Include Files | The pattern for types of files to pick up during polling. The parameter supports the "glob" syntax for using wildcard characters. | True |
| fileLocation | File Location | The file location or Object Storage bucket to poll files from. This should be a file path or the **Value** of the File Location Extendable Lookup value for the appropriate bucket (usage or event). See Referencing URIs for more information. | True |
| processingConfiguration | Processing Configuration | The **Value** of the SGG Payload Processing Configuration Extendable Lookup value | True |
| errorLocation | Error Location | The Object Storage bucket where defected portions of payload and related rejection details are stored. This should be the **Value** of the File Location Extendable Lookup value for the appropriate “error” bucket. | True |
| postProcessingAction | Post Processing Action | Defines what happens to a file after it has been processed: delete (default), archive or rename.<br>• delete: The processed file is deleted.<br>• archive: The processed file is moved/copied to the location specified for the **archiveLocation** parameter. The file is renamed by appending the run date time value to the end of file name. The extension for archived files is .gz. If this option is selected, an archive bucket should be provided for the **archiveLocation** parameter<br>• rename: The processed file is renamed using the extension specified for the **processedFileExtension** parameter, but the file is not moved. | False |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Description | Detailed Description | Required? |
|---|---|---|---|
| archiveLocation | Archive Location | The Object Storage bucket where processed files are moved to. Required if the Post Processing Action is "archive". This should be the **Value** of the File Location Extendable Lookup value for the appropriate “archive” bucket. | False |
| processedFileExtension | Processed File Extension | The text to be appended to the file name after the file has been processed. Default is "PROCESSED". | False |
| maxRaiseSize | Maximum Raise Size | This parameter defines the maximum number of files that the batch program would submit for processing on each polling cycle. Default is No Maximum. | False |
| streamOutFileLocation | Stream Out File Location | The file location or Object Storage bucket where converted initial measurement are posted for distribution to external systems. This should be a file path or the **Value** of the File Location Extendable Lookup value for the appropriate bucket (usage). See Referencing URIs for more information. Used with D1-PLPSO. | True |
| streamOutFileNamePrefix | Stream Out File Name Prefix | The prefix for files created in the Stream Out File Location. The file format will be <Stream Out File Name Prefix><batch run>_<thread>_<thread count>_<timestamp>{<_file sequence>}.json.gz. If the Stream Out File Location is set then this parameter is required. Used with D1-PLPSO. | True |
| manifestOption | Manifest Option | This parameter specifies that a manifest file be created. See Manifest File for more information. Used with D1-PLPSO. | False |
| encryptExternalKey | Encryption External Key | The key ring used for the external key provided by the recipient of the file. If populated, the batch output file will be encrypted using a runtime generated key. The generated key will be encrypted using the active public key stored on the key ring. The recipient will use their private key to decrypt the generated key so the file may be decrypted. Only external public key rings may be used. See File Encryption and Signing Keys for more information. Used with D1-PLPSO. | False |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter Name | Description | Detailed Description | Required? |
|---|---|---|---|
| encryptSignatureKey | Encryption Signature Key | The key ring for the internal key used to sign the file. If populated, the batch output file will be signed using the active private key stored on the key ring. The assumption is that the corresponding public key has been shared with the recipient. Only internal file encrypting / signing key rings may be used. See File Encryption and Signing Keys for more information. Used with D1-PLPSO. | False |
| DIST-THD-POOL | Thread Pool | Thread pool name to use if the DEFAULT thread pool is not desired. | False |

> **Note**
>
> Refer to Creating Object Storage Locations for lists of suggested Object Storage buckets for each adapter.

## Mapping Additional Fields

Measurements and device event files received from head end systems can include additional fields containing data to be imported into Oracle Utilities Smart Grid Gateway. These additional fields must be mapped to elements within the measurements and device events processed by Smart Grid Gateway.

This mapping can be performed through use of a custom Groovy script, via the “On Result List” User Exit Interceptor specified in the SGG Payload Processing Configuration Extendable lookup.

### Payload Processing Overview

This section provides an overview of the payload processing logic when mapping data in incoming measurements or device events.

#### Oracle Utilities Meter Data Framework Steps

1. The file upload process program starts reading the file.
2. The upload process creates an instance of the Payload Handler class specified on the SGG Payload Processing Configuration extendable lookup. This class handles the interaction with the User Exit Interceptor script.
3. The upload process invokes the “On Result List” User Exit Interceptor. This executes a “handler” Groovy Library Script that returns an instance of a class that implements the com.splwg.d1.domain.sgg.dg.processing.PayloadProcessingCMHandler interface. This "CM payload handler" is initialized with a stream opened for the incoming file and a business object instance of SGG Payload Processing Configuration extendable lookup.
4. The upload process parses data from the incoming document into the “Plain XML” format and transforms the Plain XML data into a set of Initial Measurements (IMDs) or/and Device Events in the “native” XML format and returns each as child nodes added to the Result List parent node.


Copyright © 2026, Oracle and/or its affiliates.

5. The upload process invokes the CM payload handler, passing the Result List parent node.
6. The CM payload handler transforms data the Result List parent node based on the logic in the Handler script. See **CM Payload Handler Steps**, below.
7. The upload process creates business object instances for all child nodes of the Result List.
8. The process steps (4-7) are repeated until the process returns NULL on step 4 (when there are no further records to process).

**CM Payload Handler Steps**

1. The CM payload handler reads the given measurement or device event inside the Result List node.
2. The CM payload handler transforms specific data elements inside the measurement or device event (such as replacing a <externalUOM> element with the standard <UOM> element) based on the logic in the handler script, and updates the Result List accordingly.

### CM Handler Payload Processing User Exit Interceptor Scripts

Mapping data is performed via a Groovy Library Script that is invoked via a user exit during payload processing. The script used in this process is a referred to as a “handler” script.

**Handler Script**

The “handler” script is responsible for transforming and mapping data elements within measurements and device events.

This script should include the gethandler method that returns an instance of a class that implements the ccom.splwg.d1.domain.sgg.processing.PayloadProcessingCMHandler interface.

To view details of the PayloadProcessingCMHandler interface in the **Javadocs Viewer**, select **Admin**, select **Implementation Tools**. Select the `com.splwg.d1.domain.sgg.dg.processing` Java package in the upper right panel, and select PayloadProcessingCMHandlerfrom the list of interfaces.

The “handler” script should be defined for the “On Result List” Payload Processing User Exit Type in the User Exit Interceptors section on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.

### Configuration Steps

Configuration of Smart Grid Gateway adapters to support mapping of non-standard data involves the following steps:

1. Create a “handler script” that transforms and maps specific data elements inside the measurement or device event (such as replacing a <externalUOM> element with the standard <UOM> element). See CM Handler Payload Processing User Exit Interceptor Scripts for more information.
2. Create an SGG Payload Processing Configuration extendable lookup value. The **Payload Handler Class Name** should reference one of the usage or event “PayloadHandler” classes listed in the Payload Handler Classes and Parameters section. See Creating SGG Payload Processing Extendable Lookup Values for more information about creating these extendable lookup values..
3. Specify the script for the “On Result List” **Payload Processing User Exit Type** in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.


Copyright © 2026, Oracle and/or its affiliates.

4. Create other objects and data used with payload processing, including:
    - Key Rings and Pairs (See Creating Key Rings and Pairs)
    - Object Storage Locations (See Creating Object Storage Locations)
    - File Storage Extendable Lookup values (See Creating File Storage Extendable Lookup Values)
    - Head End System (See Creating Head End Systems)
    - Payload Processing Batch Controls (See Creating Payload Processing Batch Controls)

- Sample Implementation

## Sample Implementation

The following sample illustrates how additional fields can be mapped into the XML format sent to Smart Grid Gateway. In this case, the unit of measure for incoming measurement data is defined by an <externalUOM>, which must be mapped to the standard <UOM> element.

**Sample Script**

The below Groovy Script will loop through a list of “D1-IMDSeeder” nodes and check the “externalUom” for a value (GD_KWH in this example), and if equal then the <UOM> element will be created with the <externalUOM> values and the <externalUOM> element will be removed.

**Script Type**: Groovy Library Script

**Step 10: Groovy Imports**

```
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import com.splwg.d1.domain.sgg.processing.PayloadProcessingCMHandler;
import com.splwg.shared.logging.Logger;
import com.splwg.shared.logging.LoggerFactory;
import com.splwg.base.support.scripting.XMLUtils;
```

**Step 20: Groovy Library Interface**

```
PayloadProcessingCMHandler getHandler()
```

**Step 30: Groovy Members**

```
public PayloadProcessingCMHandler getHandler() {
  return new CMhandler();
}

public class CMhandler implements PayloadProcessingCMHandler{

  org.w3c.dom.Document onList(org.w3c.dom.Document resultList,
org.w3c.dom.Document plainXML) {
```


Copyright © 2026, Oracle and/or its affiliates.

```
  NodeList imdList=resultList.getDocumentElement().getElementsByTagName("D1-
IMDSeeder");
  for(int i=0;i<imdList.getLength();i++){
    Element imdEl = (Element) imdList.item(i);

    //check if msrs element exists
    NodeList msrsList = imdEl.getElementsByTagName("msrs");
    if (msrsList.length > 0) {
      //get preVEE
      NodeList preVEENodeList = imdEl.getElementsByTagName("preVEE");
      Element preVEENode = (Element) preVEENodeList.item(0);
      //get externaluom
      NodeList uomList = preVEENode.getElementsByTagName("externalUOM");

      Element uomElm = uomList.length>0 ? uomList.item(0) : null;
      String uom = uomElm!=null ? uomElm.getTextContent() : "";
      //get mcIdN
      NodeList mcIdNList = preVEENode.getElementsByTagName("dvcIdN");
      if ( uom.compareTo("GD_KWH")==0 && mcIdNList.length>0 ) {
        Element externalUOMElm = resultList.createElement("uom");
        externalUOMElm.setTextContent(uom);
        //replace uom with externalUOM
        preVEENode.replaceChild(externalUOMElm, uomElm);
      }
    }
  }
  return resultList;
  }
}
```

### Sample File

Below is a sample Landis+Gyr usage file. Note that this file has a"Units" value of "GD_KWH".

```
Record Type~Record Version~Time Stamp~Premise ID~ESIID~Provisioned~Meter
ID~Purpose~Commodity~Units~Calculation
Constant~Interval~Count~FirstIntervalDateTime~Data
MEPMD01~20080519~110920100120000AM~~~~GD_LL_SN100~OK~E~GD_KWH~~000100~24~100620
10010000AM~~0.2316~~0.1416~~0.2904~~0.2628~~0.240~~0.2796~~0.2232~~0.1692~~0.0
552~~0.24096~~0.276~~0.1572~~0.0612~~0.3168~~0.132~~0.1092~~0.2532~~0.2844~~0.
198~~0.0756~~0.3084~~0.15240~~0.1788~~0.1716~~
```

### Plain XML Output

Below is the Plain XML output for the above Landis+Gyr usage file. Note that this file has a <Units> element with a value of "GD_KWH".

```
<MeterReads
xmlns="http://xmlns.oracle.com/LandisGyrUsage">
    <MeterRead>
        <Origin>IMD_INT_GD_LL_SN100_good1.lg-2020-04-13-07-42-36-092</Origin
```


Copyright © 2026, Oracle and/or its affiliates.

```
>
        <ServProvExtRefId>L+G</ServProvExtRefId>
        <RecordType>MEPMD01</RecordType>
        <RecordVersion>20080519</RecordVersion>
        <TimeStamp>11092010012000AM</TimeStamp>
        <MeterID>GD_LL_SN100</MeterID>
        <Purpose>OK</Purpose>
        <Comodity>E</Comodity>
        <Units>GD_KWH</Units>
        <Interval>000100</Interval>
        <Count>24</Count>
        <FirstIntervalDateTime>10062010010000AM</FirstIntervalDateTime>
        <Data>
            <Row v="0.2316"/>
            <Row v="0.1416"/>
            <Row v="0.2904"/>
            <Row v="0.2628"/>
            <Row v="0.240"/>
            <Row v="0.2796"/>
            <Row v="0.2232"/>
            <Row v="0.1692"/>
            <Row v="0.0552"/>
            <Row v="0.24096"/>
            <Row v="0.276"/>
            <Row v="0.1572"/>
            <Row v="0.0612"/>
            <Row v="0.3168"/>
            <Row v="0.132"/>
            <Row v="0.1092"/>
            <Row v="0.2532"/>
            <Row v="0.2844"/>
            <Row v="0.198"/>
            <Row v="0.0756"/>
            <Row v="0.3084"/>
            <Row v="0.15240"/>
            <Row v="0.1788"/>
            <Row v="0.1716"/>
        </Data>

<RawData>MEPMD01~20080519~11092010012000AM~~~~GD_LL_SN100~OK~E~GD_KWH~~000100~
24~10062010010000AM~~0.2316~~0.1416~~0.2904~~0.2628~~0.240~~0.2796~~0.2232~~0.
1692~~0.0552~~0.24096~~0.276~~0.1572~~0.0612~~0.3168~~0.132~~0.1092~~0.2532~~0
.2844~~0.198~~0.0756~~0.3084~~0.15240~~0.1788~~0.1716~~</RawData>
    </MeterRead>
</MeterReads>
```

### Result List Input

Below is the Result List input sent to the handler script. Note that this file has an <externalUOM> element with a value of "GD_KWH".

```
<?xml version="1.0" encoding="UTF-8"?>
<ResultList>
    <D1-IMDSeeder>
        <fromDateTime>2010-10-06-00.00.00</fromDateTime>
```


Copyright © 2026, Oracle and/or its affiliates.

```
        <preVEE>
            <dvcIdN>GD_LL_SN100</dvcIdN>
            <externalId>IMD_INT_GD_LL_SN100_good1.lg-2020-04-13-07-42-36-092</
externalId>
            <externalUOM>GD_KWH</externalUOM>
            <stDt>2010-10-06-00.00.00</stDt>
            <spi>3600</spi>
            <imdType>D1IL</imdType>
            <msrs>
                <mL>
                    <s>1</s><q>0.2316</q>
                </mL>
                <mL>
                    <s>2</s><q>0.1416</q>
                </mL>
                <mL>
                    <s>3</s><q>0.2904</q>
                </mL>
                <mL>
                    <s>4</s><q>0.2628</q>
                </mL>
                <mL>
                    <s>5</s><q>0.240</q>
                </mL>
                <mL>
                    <s>6</s><q>0.2796</q>
                </mL>
                <mL>
                    <s>7</s><q>0.2232</q>
                </mL>
                <mL>
                    <s>8</s><q>0.1692</q>
                </mL>
                <mL>
                    <s>9</s><q>0.0552</q>
                </mL>
                <mL>
                    <s>10</s><q>0.24096</q>
                </mL>
                <mL>
                    <s>11</s><q>0.276</q>
                </mL>
                <mL>
                    <s>12</s><q>0.1572</q>
                </mL>
                <mL>
                    <s>13</s><q>0.0612</q>
                </mL>
                <mL>
                    <s>14</s><q>0.3168</q>
                </mL>
                <mL>
                    <s>15</s><q>0.132</q>
                </mL>
                <mL>
                    <s>16</s><q>0.1092</q>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                </mL>
                <mL>
                    <s>17</s><q>0.2532</q>
                </mL>
                <mL>
                    <s>18</s><q>0.2844</q>
                </mL>
                <mL>
                    <s>19</s><q>0.198</q>
                </mL>
                <mL>
                    <s>20</s><q>0.0756</q>
                </mL>
                <mL>
                    <s>21</s><q>0.3084</q>
                </mL>
                <mL>
                    <s>22</s><q>0.15240</q>
                </mL>
                <mL>
                    <s>23</s><q>0.1788</q>
                </mL>
                <mL>
                    <s>24</s><q>0.1716</q>
                </mL>
            </msrs>
        </preVEE>
        <serviceProviderExternalId>L+G</serviceProviderExternalId>
    </D1-IMDSeeder>
</ResultList>
```

### Result List Output

Below is the Result List output sent returned by the handler script. Note that <externalUOM> element had been replaced with a <UOM> element with a value of "GD_KWH".

```
<?xml version="1.0" encoding="UTF-8"?>
<ResultList>
    <D1-IMDSeeder>
        <fromDateTime>2010-10-06-00.00.00</fromDateTime>
        <preVEE>
            <dvcIdN>GD_LL_SN100</dvcIdN>
            <externalId>IMD_INT_GD_LL_SN100_good1.lg-2020-04-13-07-42-36-092</
externalId>
            <uom>GD_KWH</uom>
            <stDt>2010-10-06-00.00.00</stDt>
            <spi>3600</spi>
            <imdType>D1IL</imdType>
            <msrs>
                <mL>
                    <s>1</s><q>0.2316</q>
                </mL>
                <mL>
                    <s>2</s><q>0.1416</q>
                </mL>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<mL>
    <s>3</s><q>0.2904</q>
</mL>
<mL>
    <s>4</s><q>0.2628</q>
</mL>
<mL>
    <s>5</s><q>0.240</q>
</mL>
<mL>
    <s>6</s><q>0.2796</q>
</mL>
<mL>
    <s>7</s><q>0.2232</q>
</mL>
<mL>
    <s>8</s><q>0.1692</q>
</mL>
<mL>
    <s>9</s><q>0.0552</q>
</mL>
<mL>
    <s>10</s><q>0.24096</q>
</mL>
<mL>
    <s>11</s><q>0.276</q>
</mL>
<mL>
    <s>12</s><q>0.1572</q>
</mL>
<mL>
    <s>13</s><q>0.0612</q>
</mL>
<mL>
    <s>14</s><q>0.3168</q>
</mL>
<mL>
    <s>15</s><q>0.132</q>
</mL>
<mL>
    <s>16</s><q>0.1092</q>
</mL>
<mL>
    <s>17</s><q>0.2532</q>
</mL>
<mL>
    <s>18</s><q>0.2844</q>
</mL>
<mL>
    <s>19</s><q>0.198</q>
</mL>
<mL>
    <s>20</s><q>0.0756</q>
</mL>
<mL>
    <s>21</s><q>0.3084</q>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                </mL>
                <mL>
                    <s>22</s><q>0.15240</q>
                </mL>
                <mL>
                    <s>23</s><q>0.1788</q>
                </mL>
                <mL>
                    <s>24</s><q>0.1716</q>
                </mL>
            </msrs>
        </preVEE>
        <serviceProviderExternalId>L+G</serviceProviderExternalId>
    </D1-IMDSeeder>
</ResultList>
```

## Itron OpenWay Payload Processing

By default, payload processing for the Itron OpenWay adapter works differently than it does for other adapters, but it can be configured to work like other adapters in which payloads are saved to a staging area in Oracle Object Storage and then processed via batch processing. For large volumes, saving to the staging area provides better performance because it allows the web service to simply drop off the payload, and the collection of saved payloads can then be processed in multiple batch processing threads.

Configuration of Itron payload processing requires the same set up and configuration as other adapters. Some specifics regarding configuration for Itron processing include the following:

- **Object Storage Locations**: Itron OpenWay payload processing uses object storage locations (including Key Rings, Key Pairs and File Storage Configuration extendable lookup values) to capture usage and event payloads or exceptions received from the head end system.
- **Head End Systems and Extendable Lookups**: Itron payload processing uses head end system and extendable lookup configuration in the same manner as other adapters.
- **SGG Payload Processing Configuration Extendable Lookup Values**: These values define payload processing details for the Itron head end system (which is referenced on the payload processing batch control).
- **Payload Processing Batch Controls**: A batch control used for payload process (based on a template batch control provided with the base package)
- **Master Configuration**: Configuration settings for Itron OpenWay payload processing, including specifying the use of a payload staging area in Object Storage. See Itron OpenWay Master Configuration for more information about configuring the system to process Itron OpenWay payloads and exceptions.

### Deferring Batch Processing

We recommend configuring a monitor process on the Pending status of the Initial Load initial measurement business objects so that the VEE and measurement creation processes can be deferred to batch processing.

Deferring processing has the following additional benefits:

1. It can ensure chronological processing for scalar reads which provides more efficient processing. Processing scalar reads out of order can result in additional work as


Copyright © 2026, Oracle and/or its affiliates.

reconciliation initial measurements will be created and processed to true up the consumption calculations.

2. It can ensure measuring components are processed in order based on their relationships. For example, if a scalar channel is configured as the check channel for an interval channel, the scalar channel will be processed first so that its data is available for sum checks and any interval gap filling required.

**Configuration**

Configuring deferred processing involves adding the “IMD Monitor – Physical Devices” Batch Control as the Monitor Process on the Pending status to the following business objects:

- D1-InitialLoadIMDInterval
- D1-InitialLoadIMDScalar

When initial measurement processing is deferred it will be necessary to schedule the “IMD Monitor – Physical Devices” Batch Control to run more frequently. It should be scheduled such that it runs with a higher thread count immediately after the Itron Scheduled Reads cycle has completed and with fewer threads throughout the rest of day to pick up any other initial measurements that might be received (such as those received via On Demand Read commands).

**Default Processing — Not Recommended**

By default, payload processing for the Itron OpenWay adapter works differently than it does for other adapters. Instead of using batch processing, Itron payloads are processed as they are received. Payloads and exceptions sent from the Itron OpenWay head end system are received via inbound web services that invoke business components that process the payloads and exceptions as they are received rather than posting them to object storage locations. See Itron OpenWay Inbound Web Services for more information about the inbound web services used to receive payloads and exceptions from the Itron OpenWay head end system. This approach is not recommended.

## Itron Field Collection System Payload Processing

Payload processing for Itron Field Collection System (FCS) involves two main stages:

- Creation of a "Meter Read Download" file to be sent to FCS detailing the meters to be read, based on Measurement Cycles, Measurement Cycle Routes, and Measurement Cycle Schedules. See Creating Meter Read Download Files below for more information about this process.
- Processing of meter read data received from FCS. Previous topics in this section include details regarding FCS payload processing.

Configuration of Itron FCS payload processing requires similar set up and configuration as other adapters. Some specifics regarding configuration for Itron FCS processing include the following:

- **Object Storage Locations**: Itron FCS payload processing uses object storage locations (including Key Rings, Key Pairs and File Storage Configuration extendable lookup values) to capture usage payloads received from the head end system.
- **Head End Systems and Extendable Lookups**: Itron payload processing uses head end system and extendable lookup configuration in the same manner as other adapters.
- **SGG Payload Processing Configuration Extendable Lookup Values**: These values define payload processing details for the Itron FCS head end system (which is referenced on the payload processing batch control).


Copyright © 2026, Oracle and/or its affiliates.

- **Payload Processing Batch Controls**: A batch control used for payload process (based on a template batch control provided with the base package)
- **Master Configuration**: Configuration settings for Itron FCS meter read download and payload processing. See Itron Field Collection Master Configuration in Master Configurations for more information.

### Characteristic Types and Lookups

Configuration of Itron FCS requires defining values for a number of characteristic types and lookups used to define specific values used with FCS. Characteristics types are referenced on the FCS master configuration, while lookups are used in both the FCS master configuration and FCS extendable lookups. Characteristic types can also be used to define characteristics on contacts, devices, measurement cycle routes, and service points (indicated in parentheses).These include the following:

Characteristic Types:

- FCS Beep Flag (Service Point)
- FCS Cost Center Code (Contact)
- FCS Instruction Location Code 1 (Service Point)
- FCS Instruction Location Code 3 (Service Point)
- FCS Optical Probe ID (Device)
- FCS Special Code (Contact)
- FCS State (Measurement Route Cycle)
- FCS Special Message (Service Point)
- FCS Tamper Count1 (Device)
- FCS Tamper Count2 (Device)
- FCS Transport Mode Code (Measurement Route Cycle)

Lookups:

- WorkSet Type (WORK_SET_TYPE)
- Wakeup Tone (WAKEUP_TONE)
- FCS Read Method (FCS_READ_METHOD_FLG)

### Creating FCS Meters

Devices used with the Itron FCS head end system must be created using the FCS - Manual Meter (D1-FCSManualMeter) business object. This means you must create at least one device type whose Device Business Object is "FCS - Manual Meter" that will be used as the device type for FCS devices.

### Creating FCS Meter Read Download Files

The FCS meter read download process is based on pre-defined measurement cycles, measurement cycle routes, and measurement cycle schedules. Refer to Measurement Cycles and Measurement Cycle Schedules for more information about settings up measurement cycles, measurement cycle routes, and measurement cycle schedules.

Once this data is in place, a series of batch processes are used to create and process meter read cycle schedule routes, create a meter read download activity, and create the meter read file to be sent to FCS.


Copyright © 2026, Oracle and/or its affiliates.

The following batch processes should be run in the following order to create FCS meter read download files.

- **FCS Create Measurement Cycle Schedule Routes (D1FCSCMS)**: This process creates Schedule Routes for Measurement Cycle Schedules whose schedule selection date is the same as the batch business date.
- **SP Measurement Cycle Schedule Route (D1-CSPSR)**: This process creates instances of the FCS SP / Measurement Cycle Schedule Route (D1-FCSSPMsrmtCycScheduleRoute) in the Pending state.

> **Note**
>
> The Business Object parameter must be set to FCS SP / Measurement Cycle Schedule Route (D1-FCSSPMsrmtCycScheduleRoute) when running this process.

- **Process SP / MC Schedule Route Records (D1-PSPSR)**: This process transitions all D1-FCSSPMsrmtCycScheduleRoute Business Object in Pending state to the Complete state.
  - The FCS Create Meter Read Download Activity (D1-FCSCMRDA) Enter algorithm creates a Meter Read Download Activity in Pending state if there is an active Install Event for the service point within the referenced Scheduled Date Time from Measurement Cycle Schedule Route.
  - The Meter Read Download Activity will be automatically transitioned to the Extracted state, where the FCS Create General Process (D1-FCSCGPE) Enter algorithm that creates records in the General Process table.

> **Note**
>
> The Business Object parameter must be set to FCS SP / Measurement Cycle Schedule Route (D1-FCSSPMsrmtCycScheduleRoute) when running this process.

- **FCS Meter Read Download Pre-Processing (D1-MRDPP)**: This process extracts the activity from the General Process record with the current batch control and batch number, and creates a new record with batch control and next batch number.
- **FCS Meter Read Download (D1FCSMRD)**: This process retrieves the records in the General Process table and related tables (Activity, Measurement Cycle Route, and so on) that match its batch control name and current batch number and generates an XML file containing corresponding meter records. This file is sent to FCS.s

## Adapter Development Kit Custom Payload Processing

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process custom payload formats, including payloads from non-supported head end systems.


- Introduction
- Custom Payload Processing Overview
- Payload Processing User Exit Interceptor Scripts
- Configuration Steps
- Sample Implementation




Copyright © 2026, Oracle and/or its affiliates.

## Introduction

As delivered, the Adapter Development Kit supports payload processing of usage and event data exported from an AMI head end system in the “native” initial measurement and device event data formats (the format of the initial measurement and device event seeder business objects).

The Adapter Development Kit can also be configured to support payload processing of data from AMI head end systems in custom and non-standard formats. This involves creation of Groovy Library Scripts to:

- Parse the payload into a “plain XML” format, and
- Transform the plain XML format into the “native” ” initial measurement and device event data formats

These scripts are executed as User Exit Interceptors configured on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Configuration Extendable Lookup Values.

## Custom Payload Processing Overview

This section provides an overview of the payload processing logic when converting incoming data into the native format.

**Oracle Utilities Meter Data Framework Steps**

1. The file upload process program starts reading the file.
2. The upload process creates an instance of the Payload Handler class. This class handles the interaction with the User Exit Interceptor scripts.
3. The upload process invokes the “On Get Parser” User Exit Interceptor. This executes a “parser” Groovy Library Script that returns an instance of a class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface.
   This “payload parser” is initialized with a stream opened for the incoming file and a business object instance of SGG Payload Processing Configuration extendable lookup.
4. The upload process invokes the “On Get Transformer” User Exit Interceptor. This executes a “transformer” Groovy Library Script that returns an instance of a class that implements `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface.
   This “payload transformer” is initialized with a string representing the origin of the payload and business object instance of SGG Payload Processing Configuration extendable lookup.
5. The upload process invokes the payload parser to parse data from the incoming document into the “Plain XML” format. See Payload Parser Steps, below.
6. The upload process invokes the payload transformer, passing the Plain XML data and a Result List parent node.
7. The payload transformer transforms the Plain XML data into a set of Initial Measurements (IMDs) or/and Device Events in the “native” XML format and returns each as child nodes added to the Result List parent node. See Payload Transformer Steps, below.
8. The upload process creates business object instances for all child nodes of the Result List.
9. The process steps (5-9) are repeated until the payload parser returns NULL on step 5 (when there are no further records to process).


Copyright © 2026, Oracle and/or its affiliates.

### Payload Parser Steps

1. The payload parser reads an input stream until it hits the logical end of portion of data (each usage or event record), which could be transformed into Plain XML.
2. The payload parser parses the data and converts into the Plain XML format.
3. The process is performed for each call from the upload process.

### Payload Transformer Steps

1. The payload transformer reads the given Plain XML.
2. The payload transformer generates a set of XML nodes representing Initial Measurements (IMDs) or Device Events. Names for nodes are taken from the provided SGG Payload Processing Configuration.
3. The payload transformer adds the generated nodes to the given Result List node.

## Payload Processing User Exit Interceptor Scripts

Conversion of data from custom and non-standard formats into the “native” format is performed via a pair of Groovy Library Scripts that are invoked via user exits during payload processing (see Custom Payload Processing Overview). The two scripts used in this process include a “parser” script, and a “transformer” script.

### Parser Script

The “parser” script is responsible for parsing the incoming data and converting it into the Plain XML format. This script should include the `getParser` method that returns an instance of a class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface.

To view details of the *PayloadParser* interface in the **Javadocs Viewer**, select **Admin**, select **Implementation Tools**. Select the `com.splwg.d1.domain.sgg.dg.processing` Java package in the upper right panel, and select *PayloadParser* from the list of interfaces.
The “parser” script should be defined for the “On Get Parser” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.

The base package includes two sample parser scripts that can be used as examples when developing custom parser scripts. See Sample Implementation.

### Transformer Script

The “transformer” script is responsible for converting data the Plain XML format into the “native” format. This script should include the `getTransformer` method that returns an instance of a class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface.

To view details of the *PayloadTransformer* interface in the **Javadocs Viewer**, select **Admin**, select **Implementation Tools**. Select the `com.splwg.d1.domain.sgg.dg.processing` Java package in the upper right panel, and select *PayloadTransformer* from the list of interfaces.

The “transformer” script should be defined for the “On Get Transformer” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing


Copyright © 2026, Oracle and/or its affiliates.

Configuration extendable lookup. See Creating SGG Payload Processing Extendable Lookup Values for more information.

The base package includes two sample transformer scripts that can be used as examples when developing custom transformer scripts. See Sample Implementation.

## Configuration Steps

Configuration of the Adapter Development Kit to support conversion of non-standard formats into the “native” format involves the following steps:

1. Create a “parser script” that converts data from the incoming format into Plain XML format.
   See Payload Processing User Exit Interceptor Scripts for more information.
2. Create a “transformer script” that converts data from the Plain XML format to the “native” format.
   See Payload Processing User Exit Interceptor Scripts for more information.
3. Create an SGG Payload Processing Configuration extendable lookup value with a **Payload Handler Class Name** of “com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits”.
   See Creating SGG Payload Processing Extendable Lookup Values for more information.
4. Specify the parser script for the “On Get Parser” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup.
   See Creating SGG Payload Processing Extendable Lookup Values for more information.
5. Specify the transformer script for the “On Get Transformer” Payload Processing User Exit Type in the **User Exit Interceptors** section on the SGG Payload Processing Configuration extendable lookup.
   See Creating SGG Payload Processing Extendable Lookup Values for more information.
6. Create other objects and data used with payload processing, including:
   - Key Rings and Pairs (See Creating Key Rings and Pairs)
   - Object Storage Locations (See Creating Object Storage Locations)
   - File Storage Extendable Lookup values (See Creating File Storage Extendable Lookup Values)
   - Head End System (See Creating Head End Systems)
   - Payload Processing Batch Controls (See Creating Payload Processing Batch Controls)

## Sample Implementation

The Smart Grid Gateway Adapter Development Kit includes sample parser and transformer scripts that demonstrate how custom payload processing works. These sample scripts include samples for processing data in comma separated values (CSV) format and in XML format.

- Comma Separated Values Format
- XML Format

## Comma Separated Values Format

This page describes the comma separated values format sample implementation.


Copyright © 2026, Oracle and/or its affiliates.

### Sample File Format

In this sample, data is stored as a comma separated values file.

**Note**: Values in this example cannot contain a literal comma character. This restriction is added for the simplicity of file parser implementation. The parsing logic can be re-implemented to avoid this restriction. It will not affect the rest of functionality.

The sample CSV file contains the following record types:

Interval Usage: Used for interval usage measurements

Device Events: Used for device events

Trail record: Used to provide the read date/time and number of interval usage and device event records in the file

### Interval Usage Format

| # | Field Name | Type | Definition |
|---|---|---|---|
| 1 | Record Type ID | Constant value “U” | Record type identifier |
| 2 | Start Date/Time | Time in the Unix Time format. | Meter read’s start date/time |
| 3 | End Date/Time | Time in the Unix Time format. | Meter read’s end date/time |
| 4 | Device Id | Arbitrary Text | Device identifier |
| 5 | Interval Duration | Numeric Integer | The time interval between readings in seconds (seconds per interval). |
| 6 | UOM | Arbitrary Text | Describes the units of the data values |
| 7 | Data Entry(s) | Numeric Floating-Point and optional Arbitrary Text separated by “:” | Each record can contain unlimited number of data entry fields. Each data entry is a set of two fields – a reading value and an optional reading status flag. Fields are separated by “:” character. The separator is not present when the reading status flag is not provided. |

### Device Event Format

| # | Field Name | Type | Definition |
|---|---|---|---|
| 1 | Record Type ID | Constant value “E” | Record type identifier |
| 2 | Event Date/Time | Time in the Unix Time format. | The date and time at which the event occurred |
| 3 | Device Id | Arbitrary Text | Device identifier |
| 4 | Event Name | Arbitrary Text | The primary identification name of the event |

### Trail Record Format

| # | Field Name | Type | Definition |
|---|---|---|---|
| 1 | Record Type ID | Constant value “T” | Record type identifier |


Copyright © 2026, Oracle and/or its affiliates.

| # | Field Name | Type | Definition |
|---|---|---|---|
| 2 | Creation Date/Time | Time in the Unix Time format. | Meter read's start date/time |
| 3 | Total Records | Numeric Integer | Displays number of records in file |

### Sample File

```
U,1,86400,DEVICE_DG_0,900,KWH,1:S,2.5,-3.99:R,4:B
E,86400,DEVICE_DG_1,Power Outage
U,2,86400,DEVICE_DG_2,900,KWH,1:S,2,3:R,4:B,5,6,7:A
E,86402,DEVICE_DG_3,Tamper attempt suspected
U,3,86400,DEVICE_DG_4,900,KWH,1:S
T,86401,5
```

### Sample Scripts

The sample implementation includes the following sample scripts designed for use with the above CSV format:

**DG_SmplPrCSV**: This sample script includes one method in the Library Interface – `getParser()`. The method returns an instance of the class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface. The class contains the following logic:

1. Parses an incoming payload in the CSV Format.
2. Transforms incoming data to Usage or Event related structures based on incoming data type
3. Returns one by one that structures in the Plain XML format

**DG_SmPrCSVCh**: This script is provided as a sample to demonstrate how to parse the CSV format in chunks (created via the `chunkSize` parameter) by utilizing a helper. This sample script includes one method in the Library Interface – `getParser()`. The method returns an instance of the class that implements the `com.splwg.d1.domain.sgg.dg.processing.ChunkProcessor` and `com.splwg.d1.domain.sgg.dg.processing.GenericPayload` interfaces.

The ChunkProcessor interface distinguishes a chunking-aware processor from the rest of implementations. A custom parser implemented in groovy can implement this interface. The `GenericPayload` interface defines methods needed to get access to a payload via specific helpers. Via this interface, a custom parser gets access to an instance of `LineReaderChunked`.

If the `chunkSize` dynamic parameter is not set in an SGG Payload Processing configuration the entire payload will be processed. See Common Parameters for more information about the `chunkSize` parameter.

The class contains the following logic:

1. Reads a line from incoming payload by utilizing the `LineReaderChunked.readLineWithinChunk()` method
2. Parses a line in the incoming payload (in the CSV Format) that was read in the previous step.


Copyright © 2026, Oracle and/or its affiliates.

3. Transforms incoming data to Usage or Event related structures based on incoming data type
4. Returns one by one that structures in the Plain XML format

**DG_SmplTrCSV**: This sample script includes one method in the Library Interface – `getTransformer()`. The method returns an instance of class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface. The class contains the following logic:

1. Transforms incoming message in the Plain XML format into the “native” format (the IMD Seeder or Device Event Seeder XML nodes).
2. Adds new XML nodes as child nodes to the given Result List.

Use the **Script** portal to view these scripts in more detail.

### Parsing

The parser script performs the following data mapping.
**Interval Usage to Plain XML Mapping**

The following table shows the mapping between fields in incoming interval data and child elements of Payload/Usage element in the Plain XML format:

| Interval Usage Field | Plain XML Element |
|---|---|
| Record Type ID | RecordType |
| Start Date/Time | StartDateTime |
| End Date/Time | EndDateTime |
| Device Id | DeviceId |
| Interval Duration | IntervalDuration |
| UOM | UOM |
| Data Entry(s) | Intervals/<br>Interval/<br>Value<br>[Status] |
| Record content from incoming file. | RawData |

### Device Event to Plain XML Mapping

The following table shows the mapping between fields in incoming interval data and child elements of Payload/Event element in the Plain XML format:

| Device Event Field | Plain XML Element |
|---|---|
| Record Type ID | RecordType |
| Event Date/Time | DateTime |
| Device Id | DeviceId |
| Event Name | Name |
| Record content from incoming file. | RawData |

### XML Schema of Plain XML Format


Copyright © 2026, Oracle and/or its affiliates.

The Plain XML consists of the Interval Usage and Device Event related elements at the same time.

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/GenericAdapter"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ga="http://xmlns.oracle.com/GenericAdapter">
  <xs:element name="Payload" type="ga:PayloadType"/>
  <xs:complexType name="PayloadType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="Usage" type="ga:UsageType"/>
        <xs:element name="Event" type="ga:EventType"/>
      </xs:choice>
      <xs:element name="RawData"  type="ga:NonEmptyString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UsageType">
    <xs:sequence>
      <xs:element name="RecordType" type="ga:RecordTypeUsageType"/>
      <xs:element name="StartDateTime" type="xs:integer"/>
      <xs:element name="EndDateTime" type="xs:integer"/>
      <xs:element name="DeviceId" type="ga:NonEmptyString"/>
      <xs:element name="IntervalDuration" type="xs:integer"/>
      <xs:element name="UOM" type="ga:NonEmptyString"/>
      <xs:element name="Intervals" type="ga:IntervalList"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RecordTypeUsageType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="U"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="IntervalList">
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="Interval" type="ga:IntervalType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IntervalType">
    <xs:sequence>
      <xs:element name="Value" type="xs:float"/>
      <xs:element name="Status" type="ga:NonEmptyString" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventType">
    <xs:sequence>
      <xs:element name="RecordType" type="ga:RecordTypeEventType"/>
      <xs:element name="DateTime" type="xs:integer"/>
      <xs:element name="DeviceId" type="ga:NonEmptyString"/>
      <xs:element name="Name" type="ga:NonEmptyString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RecordTypeEventType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="E"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="NonEmptyString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

> **Note**
>
> **Note**: A part of incoming file that been used to generate current Plain XML structure will be placed into the Payload/RawData element.

#### Transformation

The transformer script performs the following mapping.

#### Plain XML to Seeder XML Mapping

The payload transformer creates the IMD or/and Device Event Seeder XML elements as following:

**Usage Mapping**: If name of Payload/* node is equal to 'Usage' (interval usage), the script creates an XML element with a name defined in the SGG Payload Processing Configuration field “IMD Seeder BO (Interval)” and the structure according to the following mapping table:

| Plain XML Element | IMD Seeder Element | Notes |
|---|---|---|
| RecordType | N/A | |
| DeviceId | dvcIdN | |
| UOM | externalUOM | |
| IntervalDuration | spi | |
| StartDateTime | stDt | Convert to OUAF "dateTime" type |
| EndDateTime | enDt | Convert to OUAF "dateTime" type |
| Intervals/ | | |
| Interval/ | | |
| Value [Status] | msrs | Value |
| msrs/mL/q | | |
| Status -> | | |
| msrs/mL/sts/stsL/st | | |
| N/A | imdType | Const value 'D1IL' |
| | externalId | Value stored in setOrigin method. |
| | serviceProviderExternalId | Value stored in setConfiguration method. |
| RawData | rawData | If the SGG Payload Processing Configuration has the **Populate RAW** checkbox checked. |

**Device Event Mapping**: If name of Payload/* node is equal to 'Event' (device event data) create an XML element with a name defined in the SGG Payload Processing Configuration field “Device Event Seeder BO” and the structure according to the following mapping table:


Copyright © 2026, Oracle and/or its affiliates.

| Plain XML Element | Device Event Seeder Element | Note |
|---|---|---|
| RecordType | N/A | |
| DateTime | eventDateTime | Convert to OUAF "dateTime" type |
| DeviceId | externalUOM | |
| Name | externalEventName | |
| | externalSourceIdentifier | Value stored in setOrigin method. |
| | externalSenderId | Value stored in setConfiguration method. |
| RawData | rawEventInformation | If the SGG Payload Processing Configuration has the **Populate RAW** checkbox checked. |

## XML Format

This page describes the XML format sample implementation.

### Sample File Format

The file format is described in the Online IMD Upload hint on the **Online IMD and Event Upload** screen. (Select **Menu**, select **Communication**, select **Load IMDs/Events (XML)**).

### Sample File

```
<deviceList>
  <device>
    <headEnd>L&G</headEnd>
    <headEndExternalId>L+G</headEndExternalId>
    <deviceId></deviceId>
    <deviceIdentifierNumber>DEV-OUSGG-OSB-DEMO-001</deviceIdentifierNumber>
    <initialMeasurementDataList>
      <initialMeasurementData>
        <preVEE>
          <mcIdN></mcIdN>
          <uom>KWH</uom>
          <stDt>2010-05-19-00.00.00</stDt>
          <enDt>2010-05-19-00.30.00</enDt>
          <spi>900</spi>
          <msrs>
            <mL>
              <s>1</s>
              <q>0.2316</q>
            </mL>
            <mL>
              <s>2</s>
              <q>0.1416</q>
            </mL>
          </msrs>
        </preVEE>
      </initialMeasurementData>
      <initialMeasurementData>
        <initialMeasurementDataId>testValue</initialMeasurementDataId>
          <preVEE>
            <mcIdN></mcIdN>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <uom>KWH2</uom>
            <stDt>2010-05-19-00.30.00</stDt>
            <enDt>2010-05-19-01.00.00</enDt>
            <spi>900</spi>
            <msrs>
              <mL>
                <s>1</s>
                <q>1.2316</q>
              </mL>
              <mL>
                <s>2</s>
                <q>2.1416</q>
              </mL>
            </msrs>
          </preVEE>
      </initialMeasurementData>
    </initialMeasurementDataList>
  </device>
</deviceList>
```

### Sample Scripts

The sample implementation includes the following sample scripts designed for use with the above XML format:

**DG_SmplPrXML**: This sample script includes one method in the Library Interface – `getParser()`. The method returns an instance of the class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadParser` interface. The class contains the following logic:

1. Parses an incoming payload in XML format
2. Fetches information about current device
3. Breaks an incoming data into separate initial measurement data structures
4. Extends these structures with device information
5. Returns one by one these structures in the Plain XML format

**DG_SmplTrXML**: This sample script includes one method in the Library Interface – `getTransfprmer()`. The method returns an instance of class that implements the `com.splwg.d1.domain.sgg.dg.processing.PayloadTransformer` interface. The class contains the following logic:

1. Transforms incoming message in the Plain XML format into the IMD Seeder XML nodes
2. Adds new XML nodes as child nodes to the given Result List

Use the **Script** portal to view these scripts in more detail.

### Parsing

The XML parser script maps data from the above XML format into the Plain XML format.

#### XML Schema of Plain XML Format

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/GenericAdapter"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```


Copyright © 2026, Oracle and/or its affiliates.

```
  xmlns:ga="http://xmlns.oracle.com/GenericAdapter">
<xs:element name="Payload" type="ga:PayloadType"/>
<xs:complexType name="PayloadType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Usage" type="ga:UsageType"/>
      <xs:element name="Event" type="ga:EventType"/>
    </xs:choice>
    <xs:element name="RawData"  type="ga:NonEmptyString"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="UsageType">
  <xs:sequence>
    <xs:element name="RecordType" type="ga:RecordTypeUsageType"/>
    <xs:element name="StartDateTime" type="xs:integer"/>
    <xs:element name="EndDateTime" type="xs:integer"/>
    <xs:element name="DeviceId" type="ga:NonEmptyString"/>
    <xs:element name="IntervalDuration" type="xs:integer"/>
    <xs:element name="UOM" type="ga:NonEmptyString"/>
    <xs:element name="Intervals" type="ga:IntervalList"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="RecordTypeUsageType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="U"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="IntervalList">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="Interval" type="ga:IntervalType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="IntervalType">
  <xs:sequence>
    <xs:element name="Value" type="xs:float"/>
    <xs:element name="Status" type="ga:NonEmptyString" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EventType">
  <xs:sequence>
    <xs:element name="RecordType" type="ga:RecordTypeEventType"/>
    <xs:element name="DateTime" type="xs:integer"/>
    <xs:element name="DeviceId" type="ga:NonEmptyString"/>
    <xs:element name="Name" type="ga:NonEmptyString"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="RecordTypeEventType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="E"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="NonEmptyString">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
  </xs:restriction>
```


Copyright © 2026, Oracle and/or its affiliates.

```
  </xs:simpleType>
</xs:schema>
```

**Transformation**

The transformer script performs the following mapping.

**Plain XML to Seeder XML Mapping**

The payload transformer script does the following:

1. Creates an XML element with a name defined in the SGG Payload Processing Configuration field “IMD Seeder BO (Interval)” and copies all elements from the Plain XML as child nodes of new element.
2. Inserts the “externalId” element populated with value stored in `setOrigin` method.
3. Inserts the “serviceProviderExternalId” element populated with value stored in `setConfiguration` method.
4. If the SGG Payload Processing Configuration has the **Populate RAW** checkbox checked, inserts the “rawData” element populated with the textual XML representation of Plain XML element.

# Adapter Development Kit Common Information Model (CIM) Payload Processing

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process payload formats in the Common Information Model (CIM) format.

- CIM Payload Processing Introduction and Overview
- Creating CIM Payload Processing Extendable Lookup Values
- CIM Event Processing Inbound Web Service
- CIM Payload Processing Data Mapping and Extendable Lookups
- CIM Payload Processing Sample Payloads

## CIM Payload Processing Introduction and Overview

The Adapter Development Kit supports payload processing of data in the Common Information Model (CIM) format. The CIM formats and standards supported by the Adapter Development Kit are based on the *International Standard* (IEC 61968-9, Edition 2.0), available here. Reference to specific data elements from the CIM usage and event payload formats used in this section are based on this document. Refer to this document for detailed information about the CIM standard and payload formats.

Usage payloads are processed in the same manner as other formats and other adapters. Payload files are sent from a head end system to a file system (for on-premises implementations) or an Oracle Cloud Object Storage location (for cloud services), where they are processed via batch processes. See Payload Processing for more information about this process and the related configuration. Events can also be processed in this manner.

Event payload can also be processed through an Inbound Web Service, which sends the payload to a file location where it is processed via batch processing. See Event Processing Inbound Web Service for more information. This approach is most often used when receiving real-time events from a head end system.


Copyright © 2026, Oracle and/or its affiliates.

This section outlines the specific configuration necessary to process CIM-based payloads.

## Creating CIM Payload Processing Extendable Lookup Values

Payload processing of CIM usage and events requires configuration of values for the SGG Payload Processing Configuration extendable lookup. This section outlines the specific configuration for usage and event payload processing extendable lookups for CIM usage and events. See Creating SGG Payload Processing Extendable Lookup Values for more information about this extendable lookup including parameters not listed in this section.

### Usage

Extendable lookup values for CIM usage payload processing should include the following parameters:

- **Payload Handler Class Name**: com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits
- **Payload Processing Result Type**: Initial Measurements
- **IMD Seeder BO (Interval)**: IMD Seeder (D1-IMDSeeder) (Applicable only If processing interval usage)
- **IMD Seeder BO (Scalar)**: IMD Seeder (D1-IMDSeeder) (Applicable only If processing scalar usage)
- **Dynamic Parameters**: As appropriate for your implementation
- **User Exit Interceptors**:

| Payload Processing User Exit Type | Groovy Library Script | Groovy Library Method |
|---|---|---|
| On Get Parser | DG_CimUPrXML (SGG ADK Generic CIM XML Parser) | getParser |
| On Get Transformer | DG_CimUTrXML (SGG ADK Generic CIM XML Transformer) | get Transformer |

### Device Events

Extendable lookup values for CIM event payload processing should include the following parameters:

- **Payload Handler Class Name**: com.splwg.d1.domain.sgg.dg.processing.PayloadHandlerViaUserExits
- **Payload Processing Result Type**: Device Events
- **Device Event Seeder BO**: Device Event Seeder (D1-DeviceEventSeeder)
- **Dynamic Parameters**: As appropriate for your implementation
- **User Exit Interceptors**:

| Payload Processing User Exit Type | Groovy Library Script | Groovy Library Method |
|---|---|---|
| On Get Parser | DG_CimEvntPr (CIM Events Parser using XMLReader) | getParser |


Copyright © 2026, Oracle and/or its affiliates.

| Payload Processing User Exit Type | Groovy Library Script | Groovy Library Method |
|---|---|---|
| On Get Transformer | DG_CimEvntTr (CIM Events Transformer) | get Transformer |

## CIM Event Processing Inbound Web Service

Event processing can be handled via the EventMessage operation of the ADK CIM Events and Alarms (DG-CIMEvents) SOAP-based inbound web service. This approach is most often used when a head end system sends real-time events to Smart Grid Gateway.

When invoking this webservice, the event payload should be included in the web service call. Refer to the International Standard document for detailed information about the event data to be included in the web service call.

The ADK CIM Events and Alarms (DG-CIMEvents) service sends the payload to a file location where it is processed via batch processing.

The file location is defined in the **Arrived Data Save Locations** section of the ADK - Master Configuration, which includes the following parameters:

- **File Storage**: A File Storage Configuration (F1-FileStorage) extendable lookup value that defines the staging area to which the payloads will be saved. See **Creating File Storage Extendable Lookup Values** (LINK) for more information.
- **Output Bucket/Folder Name**: The bucket name (if saving to Oracle Object Storage), or a directory below the specified file storage location (if saving to a local file system).
- **Filename Generation Script**: The name of a service script used to compose the filename when saving the incoming payload. The Default Filename Generator (D1DEFFNGEN) script is used by default if not specified.
- **XSL Transformer Template**: The name of an XSL used to create the event payload from the incoming inbound message data. Select the "ADK - CIM EndDeviceEvents Output" (DG-CIMEndDvcEvtsOutput.xsl) XSL.
- **File Extension**: An extension used for the name of the outputted file. It can be specified with or without a leading dot (ex: ".txt" or "xml").

## CIM Payload Processing Data Mapping and Extendable Lookups

### Unit of Measure Mapping

Unit of measure mapping is based on the ReadingType string in the CIM payload and uses the ADK - UOM Code to Standard UOM Mapping (DG-HeadendUOMLookup) extendable lookup. Refer to the International Standard document for detailed information about the ReadingType data element.

For example, a payload with a ReadingType string of "0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0" could be mapped to the "Kilowatt Hour" unit of measure as follows:

- **Head-End Unit of Measure**: 0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0
- **Unit of Measure**: Kilowatt Hour


Copyright © 2026, Oracle and/or its affiliates.

### Interval Size Mapping

Interval size (expressed as seconds per interval or SPI) mapping is based on the measuringPeriod value extracted from the ReadingType string and uses the ADK CIM Measuring Period Lookup (DG-CIMMeasuringPeriodLookup) extendable lookup. The measuringPeriod is the third element in the ReadingType string. Refer to the International Standard document for detailed information about the measuringPeriod and ReadingType elements.

For example, the measuringPeriod in a payload with a ReadingType string of "0.0.2.4.1.1.12.0.0.0.0.0.0.0.3.72.0" is "2" (the third value in the string). This could be mapped to an SPI of 900 (15 minutes) as follows:

- **Measuring Period**: 2
- **Interval Size**: 00:15:00

### Condition Code Mapping

Condition code mapping is based on the ReadingQualityType string in the CIM payload (if present) and uses the the ADK - Interval Sts Code to Cond Map (DG-IntStsCodeToCondMapLookup) extendable lookup. Refer to the International Standard document for detailed information about the ReadingQualityType data element.

For example, a payload with a ReadingQualityType string of "3.8.8" could be mapped to the "System Estimate" condition code as follows:

- **Interval Status**: 3.8.8
- **Condition**: System Estimate

### Event Type Mapping

Event type mapping is based on the EndDeviceEventType string in the CIM payload and uses the the ADK - Device Event Mapping (DG-DeviceEventMappingLookup) extendable lookup. Refer to the International Standard document for detailed information about the EndDeviceEventType data element.

For example, a payload with a EndDeviceEventType string of "3.12.0.257"" could be mapped to the "Tamper Detected" standard event name as follows:

- **Head-End System Event Name**: 3.12.0.257
- **Standard Event Name**: Tamper Detected

## CIM Payload Processing Sample Payloads

This section provides sample CIM usage and event payloads. Refer to the International Standard document for detailed information about the data elements in these examples.

### Usage

The following example represents a meter reading containing four KWH interval values and four KW scalar readings.

```
<mr:MeterReadings
xmlns:mr="http://iec.ch/TC57/2011/MeterReadings#">
    <mr:MeterReading>
        <mr:Meter>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <mr:Names>
                <mr:name>GDDGCIM_INT_CLOUD_TESTHARNESS_01</mr:name>
                <mr:NameType>
                    <mr:description>This is an endpoint serial number</
mr:description>
                    <mr:name>meterID</mr:name>
                    <mr:NameTypeAuthority>
                        <mr:description>AssetManagementSystem</mr:description>
                        <mr:name>com.company.assets</mr:name>
                    </mr:NameTypeAuthority>
                </mr:NameType>
            </mr:Names>
        </mr:Meter>
        <mr:IntervalBlocks>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T09:30:47Z</mr:timeStamp>
                <mr:value>25.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T00:15:00Z</mr:end>
                    <mr:start>2022-02-03T00:00:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T10:00:47Z</mr:timeStamp>
                <mr:value>26.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T00:30:00Z</mr:end>
                    <mr:start>2022-02-03T00:15:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T10:30:47Z</mr:timeStamp>
                <mr:value>27.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T00:45:00Z</mr:end>
                    <mr:start>2022-02-03T00:30:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:IntervalReadings>
                <mr:reportedDateTime>2022-02-04T00:00:00Z</
mr:reportedDateTime>
                <mr:timeStamp>2022-02-03T10:30:47Z</mr:timeStamp>
                <mr:value>28.44</mr:value>
                <mr:timePeriod>
                    <mr:end>2022-02-03T01:00:00Z</mr:end>
                    <mr:start>2022-02-03T00:45:00Z</mr:start>
                </mr:timePeriod>
            </mr:IntervalReadings>
            <mr:ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </mr:IntervalBlocks>
```


Copyright © 2026, Oracle and/or its affiliates.

```
        <mr:Readings>
            <mr:timeStamp>2022-02-03T00:15:00Z</mr:timeStamp>
            <mr:value>9.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:Readings>
            <mr:timeStamp>2022-02-03T00:30:00Z</mr:timeStamp>
            <mr:value>10.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:Readings>
            <mr:timeStamp>2022-02-03T00:45:00Z</mr:timeStamp>
            <mr:value>11.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:Readings>
            <mr:timeStamp>2022-02-03T01:00:00Z</mr:timeStamp>
            <mr:value>12.0</mr:value>
            <mr:ReadingType ref="0.0.0.12.1.1.37.0.0.0.0.0.0.0.3.38.0"/>
        </mr:Readings>
        <mr:valuesInterval>
            <mr:end>2022-02-03T01:00:00Z</mr:end>
            <mr:start>2022-02-03T00:00:00Z</mr:start>
        </mr:valuesInterval>
    </mr:MeterReading>
</mr:MeterReadings>
```

### Events

The following sample represents a single "tamper" device event. Note that this sample is the event data ONLY. This is NOT an example of the data sent to the ADK CIM Events and Alarms inbound web service.

```
 <end:EndDeviceEvents
xmlns:end="http://iec.ch/TC57/2011/EndDeviceEvents#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <end:EndDeviceEvent>
        <end:createdDateTime>2022-02-10T21:54:53.404+01:00</
end:createdDateTime>
        <end:issuerID>External System 1</end:issuerID>
        <end:issuerTrackingID>123</end:issuerTrackingID>
        <end:severity>5</end:severity>
        <end:Assets>
            <end:mRID>01f8b2e5-a677-4f2c-a6c0-1e79b409c55e</end:mRID>
            <end:Names>
                <end:name>METERID</end:name>
                <end:NameType>
                    <end:name>meterID</end:name>
                    <end:NameTypeAuthority>
                        <end:name>Utility ABC</end:name>
                    </end:NameTypeAuthority>
                </end:NameType>
            </end:Names>
        </end:Assets>
        <end:EndDeviceEventType ref="3.12.0.257"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </end:EndDeviceEvent>
</end:EndDeviceEvents>
```

# Smart Meter Commands

This section provides information about implementing smart meter commands with native implementations and Oracle Utilities cloud services, including Oracle Utilities Meter Solution Cloud Service.

- Device Communication Overview
- Smart Meter Command Flows
- Sending Outbound Communications
- Receiving Inbound Communications
- Adapter Development Kit - MultiSpeak Commands
- Adapter Development Kit - Common Information Model Commands
- Adapter Development Kit - Creating Custom Commands
- Cloud Service Test Harness
- Web Service Authentication

## Device Communication Overview

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to the head-end system. This request would be for a connect/disconnect, commission/decommission, measurement data, an on-demand read, or another type of request that the head-end system supports. The head-end system receives the message, acts on the request, and returns a reply.

See Communications in the *Business User Guide*, and About Communications in the *Administrative User Guide* for basic information about command activities and communications.

### Native Implementation Command Processing

When using native implementation of Smart Grid Gateway adapters, including with Oracle Utilities cloud services, command processing works slightly differently than it does in midddleware (on-premises) implementations.

The table below provides a brief description of the communication process, and lists example objects used by the Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay. Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

| Step | Process | Example Data |
|---|---|---|
| 1. | A user initiates a remote connect command for a device.<br>A remote connect activity business object is instantiated for the command. | Activity Business Object: Remote Connect (D1-RemoteConnect) |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Process | Example Data |
|---|---|---|
| 2. | The remote connect command activity business object creates an outbound communication.<br>The specific type of outbound communication business object created is determined by the head end system (based on the processing role defined in an Enter algorithm of the “Connection Ready” status of the command activity business object’s lifecycle). | Outbound Communication Business Object: Itron - Reconnect Meter (Remote Connect)<br>(D8-ReconnectMeterDR) |
| 3. | The outbound communication creates an outbound message.<br>The specific type of outbound message created is determined by the head end system (based on the processing role defined in an Enter algorithm of the “Awaiting Response” status of the outbound communication business object’s lifecycle). | Outbound Message Type Business Object: Outbound Message for Itron connect command<br>(D8-ItronRemoteConnectOutbndMsg) |
| 4. | The outbound message is sent to the head end system via an External System, Message Sender, and Outbound Message Type.<br>The head end system sends a synchronous response to acknowledge receipt of the request. | External System: Itron OpenWayHead End System<br>Message Sender: Itron – Remote Connect (D8-Reconnect)<br>Outbound Message Type: Itron – Remote Connect (D8-RemoteConnect) |
| 5. | As the status of the command request is updated, the head end system sends periodic messages.<br>These messages are received by an Inbound Web Service which creates instances of an inbound communication business object.<br>The specific type of inbound communication business object created is determined by the Inbound Web Service. | Inbound Communication Business Object: Itron – StatusChanged<br>(D8-StatusChanged) |
| 6. | When the command request has been completed, it sends a status changed inbound communication with a “Completed” status.<br>This transitions the outbound communication business object to its “Completed” state. | Inbound Communication Business Object: Itron – StatusChanged<br>(D8-StatusChanged) |
| 7. | The remote connect command activity business object creates a “results” outbound communication.<br>The specific type of outbound communication business object created is determined by the head end system (based on the processing role defined in an Enter algorithm of the “Connection Ready” status of the command activity business object’s lifecycle). | Outbound Communication Business Object: Itron - Reconnect Meter Results<br>(D8-ReconnectMeterResultDR) |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Process | Example Data |
|---|---|---|
| 8. | The outbound communication creates an outbound message.<br>The specific type of outbound message created is determined by the head end system (based on the processing role defined in an Enter algorithm of the "Awaiting Response" status of the outbound communication business object's lifecycle). | Outbound Message Type Business Object: Outbound Message for Itron connect result command<br>(D8-ItronRemoteCntRsltOutMsg) |
| 9. | An Enter Algorithm on the "Evaluate Response" status of the outbound message business object's lifecycle evaluates the response. | Enter Algorithm: Evaluate Get Reconnect Results Response<br>(D8-EVGRCMRST) |
| 10. | The "result" outbound communication creates a completion event to update the status of the device to indicate it has been connected.<br>The specific type of completion event business object created is specified in an Enter algorithm on the "Create Completion Event" Status of the outbound communication business object's lifecycle. | Algorithm: Create Connect Completion Event from Result<br>(D8-CRCNCER)<br>Completion Event Business Object: Connect Device<br>(D1-ConnectDevice) |
| 11. | The outbound communication updates the "Connect/Disconnect Completion Flag" and the original activity business object.<br>This update is performed by an Enter algorithm on the "Completed" Status of the outbound communication business object's lifecycle. | |

In the case of commands that also request usage readings, such as On Demand Read or Scheduled Read, usage readings are sent separately via DataArrived messages separately to the DataSubscriberService Inbound Web Service, where they can be routed to Object storage for payload processing.

## Smart Meter Command Flows

This section provides information about the smart meter command flows used by supported Smart Grid Gateway adapters.

- Itron OpenWay Command Flows
- Landis+Gyr Command Flows
- Sensus Command Flows
- Silver Spring Networks Command Flows
- Silver Spring Networks MultiSpeak Command Flows
- Filtering On-Demand Read Payloads

### Itron OpenWay Command Flows


Copyright © 2026, Oracle and/or its affiliates.

The table below lists the communication flows used with each Itron OpenWay command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
  - **Result Outbound Communication** The outbound communication business object used to request the result of a command request
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Commissioning | Itron - Add Meter Definition (Commission)<br>(D8-AddMeterDefinitionsDR) | | Device Commissioning Completion Event |
| Device Decommissioning | Itron - Deregister Meter (Decommission)<br>(D8-DeregisterMeterDR) | | Device Decommissioning Completion Event |
| Remote Connect | Itron - Reconnect Meter (Remote Connect)<br>(D8-ReconnectMeterDR)<br>**Result**: Itron - Reconnect Meter Result<br>(D8-ReconnectMeterResultDR) | Itron - StatusChanged<br>(D8-StatusChanged) | Connect Device Completion Event |
| Remote Disconnect | Itron - Meter Remote Disconnect<br>(D8-DisconnectMeterDR)<br>**Result**: Itron - Disconnect Meter Result<br>(D8-DisconnectMeterResultDR) | Itron - StatusChanged<br>(D8-StatusChanged) | Disconnect Device Completion Event |
| On-Demand Read (Scalar) | Itron - Contingency Read (Scalar)<br>(D8-ReadScalarDR)<br>**Result**: Itron - On Demand Read Result<br>(D8-ReadOnDemandReadResultDR) | Itron - StatusChanged<br>(D8-StatusChanged) | Create IMD Completion Event |
| On-Demand Read (Interval) | Itron - Contingency Read (Interval)<br>(D8-ReadIntervalDR)<br>**Result**: Itron - On Demand Read Result<br>(D8-ReadOnDemandReadResultDR) | Itron - StatusChanged<br>(D8-StatusChanged) | Create IMD Completion Event |
| Scheduled Read (Scalar) | Itron - Interrogate By Group<br>(D8-InterrogateByGroupDR) | Itron - StatusChanged<br>(D8-StatusChanged) | Create IMD Completion Event |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| | **Result**: Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR) | | |
| Scheduled Read (Interval) | Itron - Interrogate By Group (D8-InterrogateByGroupDR) | Itron - StatusChanged (D8-StatusChanged) | Create IMD Completion Event |
| | **Result**: Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR) | | |
| Device Status Check | Itron - Ping By Endpoints (Status Check) (D8-PingByEndpointsDR) | Itron - StatusChanged (D8-StatusChanged) | |
| | **Result**: Itron - Ping By Endpoints Result (D8- PingByEndpointsResultDR) | | |
| Multi-Device Status Check | Itron – Ping By Endpoints Multi-Device (D8-PingByEndpointsMultiDvcDR) | Itron – StatusChanged (D8-StatusChanged) | |
| | **Result**: Itron – Ping By Endpoints Multi Dvc Result (D8-PingByEndpointsMDResultDR) | | |
| Load Check | Itron – Detect Load Side Voltage (D8-DetectLoadSideVoltgeByMtrDR) | Itron – StatusChanged (D8-StatusChanged) | |
| | **Result**: Itron – Detect Load Side Voltage Result (D8-DetLoadSideVoltageMtrRsltDR) | | |
| Read Disconnect State | Itron - Read Disconnect State (D8-ReadDisconStateByMtrDR) | Itron – StatusChanged (D8-StatusChanged) | |
| | **Result**: Itron - ReadDisconnectStateByMeter Result (D8-ReadDisconStByMtrResultDR) | | |

### Batch Processing

The Inbound Communication Wait - Monitor (D1-IBCOM) batch control is defined on the "Pending" state of the Itron - StatusChanged (D8-StatusChanged) business object. The system should be configured to run this process frequently (every few minutes) to ensure timely processing of inbound communications created by the response from the head end system.

## Landis+Gyr Command Flows


Copyright © 2026, Oracle and/or its affiliates.

The table below lists the communication flows used with each Landis+Gyr command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Commissioning | L+G Add Meter to Inventory(MultiSpeak) (D3-AddMeterToInventryMultiSpDR) | | Device Commissioning Completion Event |
| Device Decommissioning | L+G Meter Remove Notification (MultiSpeak) (D3-MtrRmvNotifMultiSpeakDR) | | Device Decommissioning Completion Event |
| Meter Retire | L+G Meter Retire Notification (D3-MeterRetireNotificationDR) | | |
| Meter Exchange | L+G Meter Exchange Notification (D3-MeterExNotificatnMultiSpDR) | | |
| Remote Connect | L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR) | Con/Discon State Change Notif-Direct Route (D3-ConnectDisconStateChgNtfDR) | Connect Device Completion Event |
| Remote Disconnect | L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR) | Con/Discon State Change Notif-Direct Route (D3-ConnectDisconStateChgNtfDR) | Disconnect Device Completion Event |
| Demand Reset | L+G Schedule Demand Reset (Multispeak) (D3-ScheduleDemandResetDR) | Reading Changed Notification Direct Route (D3-ReadingChgNotificationDR) | |
| On-Demand Read (MultiSpeak) | L+G MultiSpeak Initiate Meter Read By Meter Num Direct Route (D3-InitiateMRByMtrNbrDR) | Reading Changed Notification Direct Route (D3-ReadingChgNotificationDR) | Create IMD Completion Event |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| On-Demand Read (Interval) [CIM] | CIM Meter On Demand Read Direct Route (Interval) (D3-CIMGetLPDataDR) | CIM AsyncReply MeterReads Direct Route (D3-CIMAsyncReplyMeterReadingDR) | Create IMD Completion Event |
| On-Demand Read (Scalar) [CIM] | CIM Meter On Demand Read Direct Route (Scalar) (D3-CIMMeterOnDemandReadDR) | CIM AsyncReply MeterReads Direct Route (D3-CIMAsyncReplyMeterReadingDR) | Create IMD Completion Event |
| Device Status Check [CIM] | CIM Ping Direct Route (D3-CIMPingDR) | CIM AsyncReply MeterReads Direct Route (D3-CIMAsyncReplyMeterReadingDR) | |
| On-Demand Read (Interval) [CIM 2.0] | L+G CIM 2.0 On-Demand Read DR (Interval) (D3-CIMGetMtrReadIntervalDR) | L+G CIM 2.0 - On-Demand Read Response DR (D3-CIMMtrReadResponseDR) | Create IMD Completion Event |
| On-Demand Read (Scalar) [CIM 2.0] | L+G CIM 2.0 - On-Demand Read DR (Scalar) (D3-CIMGetMtrReadScalarDR) | L+G CIM 2.0 - On-Demand Read Response DR (D3-CIMMtrReadResponseDR) | Create IMD Completion Event |
| Device Status Check [CIM 2.0] | L+G CIM 2.0 - Ping DR (D3-CIM2PingDR) | L+G CIM 2.0 - Ping Response DR (D3-CIMPingResponseDR) | |

### Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CRRCCEDR (Create IMD Completion Event -Direct Route) algorithm on the Create Completion Events status of the D3-ReadingChgNotificationDR (L+G Reading Changed Notification Direct Route) business object.

The D3-VALDVCFDR (Check if Device has any Filtering errors Direct Route) algorithm on the Validate status of the D3-ReadingChgNotificationDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the same parameter on the D3-CRRCCEDR algorithm. The “Opt Out Of Filtering” parameter should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

### CIM Commands


Copyright © 2026, Oracle and/or its affiliates.

For CIM-based on-demand read commands, filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D3-CRCIMCE (Create CIM Completion Event Direct Route) algorithm on the Create Completion Events status of the D3-CIMAsyncReplyMeterReadingDR (CIM AsyncReply MeterReads Direct Route) business object.

The D3-VALCCIMCE (Check if Device has Filtering errors CIM Completion Event Direct Route) algorithm on the Validate status of the D3-CIMAsyncReplyMeterReadingDR (CIM AsyncReply MeterReads Direct Route) business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” parameter on this algorithm must match the value of the parameter on the D3-CRCIMCE algorithm. The parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

## Sensus Command Flows

The table below lists the communication flows used with each Sensus command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Commissioning (MultiSpeak) | Sensus -Meter Add Notification Direct Route (D6-MeterAddNotificationDR) | | Device Commissioning Completion Event |
| Device Decommissioning (MultiSpeak) | Sensus -Meter Remove Notification Direct Route (D6-MeterRemoveNotificationDR) | | Device Decommissioning Completion Event |
| Remote Connect (MultiSpeak) | Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR) | Sensus Conn/Discon State Chg Direct Route (D6-ConnectDisconStateChgNtfDR) | Connect Device Completion Event |
| Remote Disconnect (MultiSpeak) | Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR) | Sensus Conn/Discon State Chg Direct Route (D6-ConnectDisconStateChgNtfDR) | Disconnect Device Completion Event |
| On-Demand Read (MultiSpeak) | Sensus - Initiate Meter Read By Meter ID (D6-InitiateMeterByMeterIdDR) | Sensus - Reading Changed Notification (D6-ReadingChgNotificationDR) | Create IMD Completion Event |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Status Check (MultiSpeak) | Sensus Initiate Outage Detection DR (D6-InitiateOutageDetectionDR) | Sensus Outage Detect Event Notif Direct Route (D6-ODEventNotificationDR) | |

**Unit of Measure Filtering for On-Demand Reads**

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D6-CCOMPEVDR (Create IMD Completion Event Direct Route) algorithm on the Create Completion Events status of the D6-ReadingChgNotificationDR (Sensus - Reading Changed Notification DR) business object.

The D6-VALDVCFDR (Check if Device has any Filtering errors Direct Route) algorithm on the Validate status of the D6-ReadingChgNotificationDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” and "Xpath to External UOM" parameters on this algorithm must match the values of the same parameters on the D6-CCOMPEVDR algorithm. The “Opt Out Of Filtering” parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

## Silver Spring Networks Command Flows

The table below lists the communication flows used with each Silver Spring Networks command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
  - **Get Status Outbound Communication**: The outbound communication business object used to request the status of a command request
  - **Result Outbound Communication**: The outbound communication business object used to request the result of a command request
- **Outbound Messages**: The outbound message business objects for one or more outbound messages created by the outbound communication. These should be configured on the Silver Spring Networks external system.
- **Inbound Communications**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Outbound Messages | Completion Event |
|---|---|---|---|
| Device Commissioning (see Note below) | SSN - Replace Location (D7-ReplaceLocationDR) | **Find Device**:<br>SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg)<br>**Replace Location**:<br>SSN - Replace Location (Comm) OB Message (D7-ReplaceLocationOBMsg) | Device Commissioning Completion Event |
| Device Decommissioning | SSN - Replace Device At Location (Decomm) (D7-ReplDeviceAtLocForDecommDR) | **Find Device**:<br>SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg)<br>**Replace Device at Location**:<br>SSN - Replace Device At Loc O/B Message: (D7-ReplaceDeviceAtLocOBMsg) | Device Decommissioning Completion Event |
| Remote Connect | SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP) | **Find Device**:<br>SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg)<br>**Connect Disconnect**:<br>SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg) | Connect Device Completion Event |
| | **Get Status**: SSN - Get Job Status for Polling (D7-GetJobStatusDRP) | SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg) | |
| | **Result**: SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR) | SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg) | |
| Remote Disconnect | SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP) | **Find Device**:<br>SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg)<br>**Connect Disconnect**:<br>SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg) | Disconnect Device Completion Event |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Outbound Messages | Completion Event |
|---|---|---|---|
| | **Get Status**: SSN - Get Job Status for Polling (D7-GetJobStatusDRP) | SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg) | |
| | **Result**: SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR) | SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg) | |
| On-Demand Read (Scalar) | SSN - Add Mtr Read Job (Scalar) - Polling (D7-AddMeterReadJobScalarDRP) | **Find Device**:<br>SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg)<br>**Add Meter Read Job Scalar**: SSN - Add Meter Read Job Scalar OB Msg Type<br>(D7-AddMeterReadJobScalarOBMsg) | Create IMD Completion Event |
| | **Get Status**: SSN - Get Job Status for Polling (D7-GetJobStatusDRP) | SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg) | |
| | **Result**: SSN - Meter Read Results (Scalar)<br>(D7-MeterReadResultsScalar) | SSN - GetMeterReadResults(Scalar) By JobId<br>(D7-GetSclrMtrRdRstByJobIdOBMsg) | |
| On-Demand Read (Interval) | SSN - Add Mtr Read Job (Interval) - Polling (D7-AddMeterReadJobIntervalDRP) | **Find Device**:<br>SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg)<br>**Add Meter Read Job Interval**: SSN - AddMeterReadJob Interval OB Msg Type<br>(D7-AddMeterReadJobIntervalOBMsg) | Create IMD Completion Event |
| | **Get Status**: SSN - Get Job Status for Polling (D7-GetJobStatusDRP) | SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg) | |
| | **Result**: SSN - Meter Read Results (Interval)<br>(D7-MeterReadResultsInterval) | SSN - GetMeterReadResults(Interval) By JobId<br>(D7-GetIntMtrRdRsltByJobIdOBMsg) | |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Outbound Messages | Completion Event |
|---|---|---|---|
| Device Status Check | SSN - Add Ping Job - Polling (D7-AddPingJobDRP) | **Find Device**:<br>SSN - Find Device Outbound Message (D7-FindDeviceOutboundMsg)<br>**Add Ping Job**:<br>SSN - Add Ping Job OB Message (D7-AddPingJobOBMsg) | |
| | **Get Status**: SSN - Get Job Status for Polling (D7-GetJobStatusDRP) | SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg) | |
| | **Result**: SSN - Ping Results (D7-PingResultsDR) | SSN - Get Ping Results O/B Message (D7-GetPingResultsOBMsg) | |

> **Note**
>
> Service points must be provisioned in the Silver Spring Networks (Itron AMM) system before the Device Commissioning command can be used.

### Configuring Communication Types for Job Status Polling

The following outbound communication business objects (from the table above) all have corresponding communication types that must be configured before outbound communications based on those business objects can be created. The table below lists the communication type business objects used with each.

| Outbound Communication Business Object | Communication Type Business Object |
|---|---|
| SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP) | SSN - Add Remote Prov Job Polling DR Type (D7-AddRemoteProvDRPType) |
| SSN - Add Mtr Read Job (Scalar) - Polling (D7-AddMeterReadJobScalarDRP) | SSN - Add Meter Read Job (Scalar) Polling DR Type (D7-AddMeterReadJobScalrDRPType) |
| SSN - Add Mtr Read Job (Interval) - Polling (D7-AddMeterReadJobIntervalDRP) | SSN - Add Meter Read Job (Interval) Polling DR Type (D7-AddMeterReadJobIntDRPType) |
| SSN - Add Ping Job - Polling (D7-AddPingJobDRP) | SSN - Add Ping Job Polling DR Type (D7-AddPingJobDRPType) |
| Get Job Status for Polling (D7-GetJobStatusDRP) | SSN - Get Job Status Polling DR Type (D7-GetJobStatusDRPType) |

Refer to **Communication Types** for more information about communication types.

### Get Job Status Polling Parameters and Polling Modes

The **Get Job Status Polling** parameters on these communication types are used to define how polling works with outbound communications of the corresponding type. These include **Retry Frequency** parameters (the minimum amount of time (expressed as hours:minutes:seconds) the system will wait between polling iterations for jobs created by communications of this type) and **Maximum Retries** parameters (the maximum number of


Copyright © 2026, Oracle and/or its affiliates.

times the system will try to poll for the status of jobs created by communications of this type), which must be configured in order to use these outbound communications.

There are two modes of polling:

- **Initial** polling occurs immediately and within the same transaction as any previous portion of the command. This setting is useful when the head end is expected to quickly complete the job (within a few seconds). To use Initial polling, define values for the **Initial Polling Retry Frequency** and **Initial Polling Maximum Retries** parameters.
- **Out of Process** polling occurs within its own transaction triggered either separately by a user or by a batch process. This mode is useful when the head end system is likely to take more than a few seconds to complete the job. The **Polling Retry Frequency** and **Polling Maximum Retries** parameters are used with Out of Process polling. If these parameters are not configured on the communication type, the values defined in the **SSN - Get Job Status Configuration** section of the Silver Spring Networks Version Master Configuration are used in their place.

Refer to the **Get Job Status Polling** embedded help on the above communication types for more information about polling modes and parameters.

### Polling and Batch Processing

When a communication enters the "Polling Ready" state, it polls the head end system via a Get Status Outbound Communication. If the initial polling parameters have been exceeded (or the initial polling parameters were not defined), the communication transitions to the "Polling in Progress" state. Communications in this state can be transitioned using the Outbound Communication Polling - Monitor (D7-OCPL) batch process (or the **Poll Job Status** button in the **Outbound Communication** zone in the **Communication** portal.)

If using the Outbound Communication Polling - Monitor (D7-OCPL) batch process, the system should be configured to run the process frequently (every few minutes).

Polling and the creation of Get Job Status outbound communications requires the following "Get Job Status" processing method be defined for the Silver Spring Networks head end system:

- **Processing Role**: Get Job Status
- **Default Business Object**: SSN - Get Job Status for Polling (D7-GetJobStatusDRP)

See **Silver Spring Networks Processing Methods** for more information about processing methods used with Silver Spring Networks.

### Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the D7-CCOMPEVSR (SSN - Create Completion Event Scalar) algorithm on the Create Completion Events status of the D7-MeterReadResultsScalar (SSN - Meter Read Results (Scalar)) business object (for scalar on-demand read commands) and/or on the D7-CCEVTINDR (SSN - Create Completion Event Interval) algorithm on the Create Completion Events status of the D7-MeterReadResultsInterval (SSN - Meter Read Results (Interval)) business object (for interval on-demand read commands).

When filtering is enabled, the “Filter if invalid external MC identifier” parameter on the D7-CCOMPEVSR and D7-CCEVTINDR algorithms controls what should happen if a measuring


Copyright © 2026, Oracle and/or its affiliates.

component cannot be found using the MC Identifier Number sent from the head end system. When set to 'Yes' and no measuring component is found, the initial measurement will be filtered. When set to 'No' and no measuring component is found, the system will attempt to find the measuring component using the UOM and TOU. The value of the “Filter if invalid external MC identifier” on the D7-CCOMPEVSR and D7-CCEVTINDR algorithms must match the value of the “Error if invalid external MC identifier” parameter on the D1-DER-SPRMC algorithm on the IMD Seeder (D1-IMDSeeder) business object. The parameters should both be set to either ‘yes’ or ‘no’.

### Deprecated Outbound Communications

The outbound communication business objects listed in the table above replace the following outbound communication business objects, which will be deprecated in a future release.

| Command | Deprecated Outbound Communication Business Object |
|---|---|
| Remote Connect/Remote Disconnect | SSN - Connect or Disconnect (D7-ConnectDisconnectDR) |
| On-Demand Read (Scalar) | SSN - Add Meter Read Job (Scalar) (D7-AddMeterReadJobScalarDR) |
| On-Demand Read (Interval) | SSN - Add Meter Read Job (Interval) (D7-AddMeterReadJobIntervalDR) |
| Device Status Check | SSN - Add Ping Job (D7-AddPingJobDR) |

## Silver Spring Networks MultiSpeak Command Flows

The table below lists the communication flows used with each Silver Spring Networks MultiSpeak command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Commissioning (MultiSpeak) | SSN - MultiSpeak Meter Add Notification DR (D7-MSPMeterAddNotificationDR) | | Device Commissioning Completion Event |
| Device Decommissioning (MultiSpeak) | SSN - MSP Meter Remove Notification DR (D7-MSPMeterRemNotifyDR) | | Device Decommissioning Completion Event |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Remote Connect (MultiSpeak) | SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR) | SSN - MSP Conn/Discon State Change DR (D7-MSPConDisconStaChgNtfDR) | Connect Device Completion Event |
| Remote Disconnect (MultiSpeak) | SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR) | SSN - MSP Conn/Discon State Change DR (D7-MSPConDisconStaChgNtfDR) | Disconnect Device Completion Event |
| On-Demand Read (MultiSpeak) | SSN - MSP Initiate Read By Meter ID DR (D7-MSPInitMtrByMtrIdDR) | SSN - MSP Reading Changed Notification DR (D7-MSPReadingChgNotifyDR) | Create IMD Completion Event |
| Device Status Check (MultiSpeak) | SSN - MSP Initiate Outage Detection DR (D7-MSPInitOutageDetectDR) | SSN - MSP OD Event Notification DR (D7-MSPODEventNotificationDR) | |

**Unit of Measure Filtering for On-Demand Reads**

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match a configured UOM/TOU/SQI combination are processed. All others are filtered out.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the MSP Create IMD Completion Event DR (D1-MSCCEDR) algorithm on the Create Completion Events status of the MultiSpeak Reading Changed Notification DR (D1-MSPReadingChgNotifyDR) business object.

The MSP Check if Device has any Filtering errors DR (D1-MSVADVFDR) algorithm on the Validate status of the D1-MSPReadingChgNotifyDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” and "Xpath to External UOM" parameters on this algorithm must match the values of the same parameters on the D1-MSCCEDR algorithm. The “Opt Out Of Filtering” parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

## Filtering On-Demand Read Payloads

By default, the On-Demand Read command is initiated for a specific meter (device) and returns readings (measurement data) for each register or channel (measuring component) on the meter. However, in some circumstances, an implementation may not want measurements returned for every channel. For instance, they may only want or need billing-related data, or only a subset of the data recorded by the meter.

Measurement payloads received via On-Demand Read commands can be filtered based on unit of measure (UOM) by adding custom logic to the XLS files used to send and receive On-Demand Read command.


Copyright © 2026, Oracle and/or its affiliates.

This filtering can be implemented by editing the appropriate XSL to include a variable containing a list of "approved UOMs" list and logic that filters the payload based on the "approved list" variable.

### Examples

The lines noted by “<!-- List of approved UOMS -->” and “<!-- UOM Filtering here -->” in the following excerpts from Landis+Gyr XSL files illustrate how the files can be edited to apply UOM filtering.

#### L+G Reading Changed Notification Request (D3-ReadingChgNotifRequest.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet exclude-result-prefixes="soap11 soap12 ns1" xmlns:ns1="http://
www.multispeak.org/Version_3.0" xmlns:soap12="http://www.w3.org/2003/05/soap-
envelope" xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:import href="D3-MultiSpeakMsgHeader.xsl"/>
    <!-- List of approved UOMS -->
    <xsl:variable name="Approved_UOMs">|kWh|</xsl:variable>
  <xsl:template match="soap11:Envelope">
    <xsl:apply-templates select="//ns1:ReadingChangedNotification"/>
    ...
<xsl:template match="ns1:readingValues">
  <xsl:for-each select="./ns1:readingValue">
    <readingValue>
      <extensions>
        <xsl:value-of select="./ns1:extensions"/>
      </extensions>
      <extensionsList>
        <xsl:apply-templates select="./ns1:extensionsList"/>
      </extensionsList>
      <units>
        <xsl:value-of select="./ns1:units"/>
      </units>
      <value>
        <xsl:value-of select="./ns1:value"/>
      </value>
      <!-- UOM Filtering here -->
      <xsl:if test="contains($Approved_UOMs, concat('|', ./ns1:name , '|'))">
        <name>
          <xsl:value-of select="./ns1:name"/>
        </name>
      </xsl:if>
      <dateTime>
        <xsl:value-of select="./ns1:dateTime"/>
      </dateTime>
    </readingValue>
  </xsl:for-each>
</xsl:template>
...
```


Copyright © 2026, Oracle and/or its affiliates.

### SGG L+G CIM AsyncReply MeterReadings Request XSL (D3-CIMMeterReadingRequest.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet exclude-result-prefixes="xsl env sec msg mtr ede xsi"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ede="http://
iec.ch/TC57/2009/EndDeviceEvents#" xmlns:mtr="http://iec.ch/TC57/2009/
MeterReadings#" xmlns:msg="http://www.iec.ch/TC57/2008/schema/message"
xmlns:sec="http://www.landisgyr.com/iec61968/2010/03" xmlns:env="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:xsl="http://www.w3.org/1999/XSL/
Transform" version="1.0">
<xsl:output indent="yes" encoding="utf-8" method="xml"/>
  <!-- List of approved UOMS -->
  <xsl:variable name="Approved_UOMs">|kWh|</xsl:variable>
<xsl:template match="env:Envelope">
  <xsl:apply-templates select="./env:Body/msg:ResponseMessage"/>
</xsl:template>
...

<xsl:template match="mtr:Readings">
  <xsl:if test="string-length(./*) > 0">
    <Readings>
      <timeStamp>
        <xsl:value-of select="./mtr:timeStamp"/>
      </timeStamp>
      <xsl:apply-templates select="./mtr:value" mode="copyif"/>
      <xsl:for-each select="./ReadingQualities[string-length(./mtr:quality) >
0]">
        <ReadingQualities>
          <quality>
            <xsl:value-of select="./mtr:quality"/>
          </quality>
        </ReadingQualities>
      </xsl:for-each>
      <xsl:if test="string-length(./mtr:ReadingType/@ref) > 0">
        <!-- UOM Filtering here -->
        <xsl:if test="contains($Approved_UOMs, concat('|', ./mtr:ReadingType/
@ref, '|'))">
          <ReadingTypeRef>
            <xsl:value-of select="./mtr:ReadingType/@ref"/>
          </ReadingTypeRef>
        </xsl:if>
      </xsl:if>
    </Readings>
  </xsl:if>
</xsl:template>
...
```

### Head End System XSL Files

The table below lists the XSL files that can be used to filter payloads using this approach for each head end system.

XSL files can be edited using the **Managed Content** portal.


Copyright © 2026, Oracle and/or its affiliates.

| Head End System | XSL | Where Used |
|---|---|---|
| Landis+Gyr | D3-CIMMeterReadingRequest.xsl | Inbound Web Service (CIM Service Callback) |
| Landis+Gyr | D3-ReadingChgNotifRequest.xsl | Inbound Web Service (Reading Changed Notification (Callback)) |
| Sensus | D6-ReadingChgNotifRequest.xsl | Inbound Web Service (Sensus MultiSpeak CB_Server Interface) |
| Silver Spring Networks | D7-GetMtrRdRsByJobIDResp.xsl | External System Message XSL (provided by implementation) |

## Sending Outbound Communications

Sending outbound communications to a head end system involves configuring the following entities:

- **Outbound Message Types**: Define how outbound messages of a specific type are handled. Outbound message types are based on a specific outbound message business object that defines the data structure of the message. See Defining Outbound Message Types in the *Administrative User Guide* and Creating Outbound Message Types below for more information.
- **Message Senders**: Define details for sending messages to external systems. See Message Senders in the *Administrative User Guide* for more information.
- **External System**: Defines a set of outbound message types that can be sent to the head end system. See External Systems in the *Administrative User Guide* for more information.
- **Head End System**: Defines processing rules for the head end system, such as the specific activities and communications created for smart meter commands. See Head End Systems and Processing Methods in the *Administrative User Guide* and Configuration Processing Methods below for more information.
- Creating Outbound Message Types
- Creating Message Senders
- Creating an External System
- Configuring Processing Methods

## Creating Outbound Message Types

Outbound message types must be configured for each command you plan to use in your implementation. The base package includes a set of outbound message types that can be used. The tables below lists the outbound message types provided for each Smart Grid Gateway adapter.

### Itron OpenWay Outbound Message Types

This table lists the outbound message types used with the Itron OpenWay adapter, along with the business objects used with each.


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Message Type / Business Object | "Result" Outbound Message Type / Business Object |
|---|---|---|
| Device Commissioning | Itron - Add Meter Definitions O/B Msg Type (D8-AMDOBM)<br>Itron - Commissioning (D8-ItronCommissionOutboundMsg) | N/A |
| Device Decommissioning | Itron - Deregister Meter O/B Msg Type (D8-DREGOBM)<br>Itron - Decommissioning (D8-ItronDecommissionOutbndMsg) | N/A |
| Remote Connect | Itron - Reconnect Meter O/B Msg Type (D8-RCNOBM)<br>Itron - Remote Connect (D8-ItronRemoteConnectOutbndMsg) | Itron - Get Reconnect Meter Result O/B Msg Type (D8-GRCNOBM)<br>Itron - Get Remote Connect Results (D8-ItronRemoteCntRsltOutMsg) |
| Remote Disconnect | Itron - Disconnect Meter O/B Msg Type (D8-DISCOMOBM)<br>Itron - Disconnect (D8-ItronRemoteDisconnectOutMsg) | Iron - Get Disconnect Meter Result O/B Msg Type (D8-GDMROBM)<br>Itron - Get Disconnect Result (D8-ItronRemoteDcntRsltOutMsg) |
| On-Demand Read (Scalar) | Itron - Contingency Read O/B Msg Type (D8-CRDOBM)<br>Itron - Contingency Read (D8-ItronOnDemandReadOutbndMsg) | Itron - Get Contingency Read Results O/B Msg Type (D8-GCRDOBM)<br>Itron - Get Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg) |
| On-Demand Read (Interval) | Itron - Contingency Read O/B Msg Type (D8-CRDOBM)<br>Itron - Contingency Read (D8-ItronOnDemandReadOutbndMsg) | Itron - Get Contingency Read Results O/B Msg Type (D8-GCRDOBM)<br>Itron - Get Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg) |
| Scheduled Read (Scalar) | Itron - Scheduled Read O/B Msg Type (D8-SROBM)<br>Itron - Schedule Read (D8-ItronInterrogateByGrpOBMsg) | Itron - Get Scheduled Read Result O/B Msg Type (D8-GSRROBM)<br>Itron - Get Schedule Read Result (D8-ItronInterrogateByGrpRsltOB) |
| Scheduled Read (Interval) | Itron - Scheduled Read O/B Msg Type (D8-SROBM)<br>Itron - Schedule Read (D8-ItronInterrogateByGrpOBMsg) | Itron - Get Scheduled Read Result O/B Msg Type (D8-GSRROBM)<br>Itron - Get Schedule Read Result (D8-ItronInterrogateByGrpRsltOB) |
| Device Status Check | Itron - Ping By Endpoints O/B Msg Type (D8-DSCOBM)<br>Itron - Device Status Check (D8-ItronDvcStatusChkOutbndMsg) | Itron - Get Ping By Endpoints Result O/B Msg Type (D8-GDSCROBM)<br>Itron - Get Device Status Check Result (D8-ItronDvcStatusChkRstlOBMsg) |
| Multi-Device Status Check | Itron - Multi Ping By Endpoints O/B Msg Type (D8-MDSCOBM)<br>Itron - Multi Device Status Check (D8-ItronDvcStChkMtDvcOutbndMsg) | Itron - Get Multi Ping By Endpoints Result O/B Msg Type (D8-GMDSCOBM)<br>Itron - Get Multi Device Status Check Result (D8-ItronDvcStChkMtDvcRsltOBMsg) |
| Load Check | Itron - Detect Load Side Voltage O/B Msg Type (D8-DLSVOBM)<br>Itron - Detect Load Side Voltage<br>(D8-ItronDetectLoadSideVolOBMsg) | Itron - Get Detect Load Side Voltage Results O/B Msg Type (D8-GDLSVOBM)<br>Itron - Get Detect Load Side Voltage Result<br>(D8-ItronDetLSVoltMtrRsltOutMsg) |


Copyright © 2026, Oracle and/or its affiliates.

### Landis+Gyr Outbound Message Types

This table lists the outbound message types used with the Landis+Gyr adapter, along with the business objects used with each.

| Command | Outbound Message Type / Business Object |
|---|---|
| Device Commissioning | Add Meter to Inventory (L+G) (D3-AMIOMTYP)<br>LG - Add Meter To Inventory (D3-AddMeterToInventoryOBMsg) |
| Device Decommissioning | Notify Remove Meter (L+G) (D3-MRENOMTYP)<br>LG - Meter Remove Notification (D3-MeterRemoveNotifOBMsg) |
| Meter Retire | Notify Retire Meter (L+G) (D3-MRNOMTYP)<br>LG - Meter Retire Notification (D3-MeterRetireNotifOBMsg) |
| Meter Exchange | Notify Exchange Meter (L+G) (D3-MENOBMTYP)<br>LG -Meter Exchange Notification (D3-MeterExchNotificatnOBMsg) |
| Remote Connect | Initiate Connect Disconnect by Meter Number (L+G) (D3-INCODIOBM)<br>LG - Initiate Connect Disconnect (D3-InitConnectDisconnectOBMsg) |
| Remote Disconnect | Initiate Connect Disconnect by Meter Number (L+G) (D3-INCODIOBM)<br>LG - Initiate Connect Disconnect (D3-InitConnectDisconnectOBMsg) |
| Demand Reset | Schedule Demand Reset (L+G) (D3-SDROMTYP)<br>LG - Schedule Demand Reset (D3-ScheduleDemandResetOBMsg) |
| On-Demand Read (MultiSpeak) | Initiate Read by Meter Number (L+G) (D3-INMRMNOBM)<br>LG - Initiate Meter Read By Meter Number (D3-InitMtrReadByMtrNumOBMsg) |
| On-Demand Read (Interval) [CIM] | Initiate On-Demand Read (CIM L+G) (D3-CIMODR)<br>LG - CIM AMIRequest GetMeterReads (D3-CIMAMIRequestGetMtrRdOBMsg) |
| On-Demand Read (Scalar) [CIM] | Initiate On-Demand Read (CIM L+G) (D3-CIMODR)<br>LG - CIM AMIRequest GetMeterReads (D3-CIMAMIRequestGetMtrRdOBMsg) |
| Device Status Check [CIM] | Initiate Device Status Check (CIM L+G) (D3-CIMDSC)<br>LG - CIM AMI Request Get Meter Reads (D3-CIMAMIRequestGetMtrRdOBMsg) |
| On-Demand Read (Interval) [CIM 2.0] | L+G CIM - 2.0 Get MeterReadings (D3-CIMGMR)<br>L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg) |
| On-Demand Read (Scalar) [CIM 2.0] | L+G CIM - 2.0 Get MeterReadings (D3-CIMGMR)<br>L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg) |
| Device Status Check [CIM 2.0] | L+G CIM - 2.0 Get MeterReadings (D3-CIMGMR)<br>L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg) |


Copyright © 2026, Oracle and/or its affiliates.

### Sensus Outbound Message Types

This table lists the outbound message types used with the Sensus adapter, along with the business objects used with each.

| Command | Outbound Message Type / Business Object |
|---|---|
| Device Commissioning | Notify Add Meter (Sensus) (D6-SMANOMTYP)<br>Sensus - Meter Add Notification (D6-MeterAddNotificationOBMsg) |
| Device Decommissioning | Notify Remove Meter (Sensus) (D6-SMRNOMTY)<br>Sensus -Meter Remove Notification OB Msg (D6-MeterRemoveNotificatnOBMsg) |
| Remote Connect | Initiate Connect Disconnect by Meter Number (Sensus) (D6-INCODIOBM)<br>Sensus - Initiate Connect Disconnect (D6-InitConnectDisconnectOBMsg) |
| Remote Disconnect | Initiate Connect Disconnect by Meter Number (Sensus) (D6-INCODIOBM)<br>Sensus - Initiate Connect Disconnect (D6-InitConnectDisconnectOBMsg) |
| On-Demand Read (MultiSpeak) | Initiate Read by Meter Id (Sensus) (D6-SIMROMTY)<br>Sensus - Initiate Meter Read By Meter ID (D6-InitMtrRdMtrIdOBMsg) |
| Device Status Check | Initiate Outage Detection (Sensus) (D6-SIODTOMTY)<br>Sensus - Initiate Outage Detection (D6-InitOutageDetectionOBMsg) |

### Silver Spring Networks Outbound Message Types

This table lists the outbound message types used with the Silver Spring Networks adapter, along with the business objects used with each.

| Command | Outbound Message Type / Business Object | "Result" Outbound Message Type / Business Object |
|---|---|---|
| Device Commissioning | Replace Location (Commission SSN) (D7-RLOCOBMTY)<br>SSN - Replace Location Command (D7-ReplaceLocationOBMsg) | N/A |
| Device Decommissioning | Replace Device (Decommission SSN) (D7-RDVCOBMTY)<br>SSN - Replace Device At Locaction (D7-ReplaceDeviceAtLocOBMsg) | N/A |
| Remote Connect | Connect Disconnect Device (SSN) (D7-CDCOBMTYP)<br>SSN - Connect Disconnect (D7-ConnectDisconnectOBMsg) | Get Connect Disconnect Result (SSN) (D7-GCDROBMTY)<br>SSN - Get Connect Disconnect Result (D7-GetCntDiscntResultOBMsg) |
| Remote Disconnect | Connect Disconnect Device (SSN) (D7-CDCOBMTYP)<br>SSN - Connect Disconnect (D7-ConnectDisconnectOBMsg) | Get Connect Disconnect Result (SSN) (D7-GCDROBMTY)<br>SSN - Get Connect Disconnect Result (D7-GetCntDiscntResultOBMsg) |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Message Type / Business Object | "Result" Outbound Message Type / Business Object |
|---|---|---|
| On-Demand Read (Scalar) | Add Scalar Read Job (SSN) (D7-AMRJSCOBT) | Get Scalar Read Results by Job Id (SSN) (D7-GMRRBJOMT) |
| | SSN - Add Meter Read Job Scalar (D7-AddMeterReadJobScalarOBMsg ) | SSN - Get Meter Read Results(Scalar) (D7-GetSclrMtrRdRstByJobIdOBMsg ) |
| On-Demand Read (Interval) | Add Interval Read Job (SSN) (D7-AMRJINOBT) | Get Interval Read Results by Job Id (SSN) (D7-GIMRRJOMT) |
| | SSN - Add Meter Read Job Interval (D7-AddMtrReadJobIntervalOBMsg ) | SSN - Get Meter Read Results(Interval) (D7-GetIntMtrRdRsltByJobIdOBMsg ) |
| Device Status Check | Add Meter Ping Job (SSN) (D7-APJOBM) | Get Device Ping Job Results (SSN) (D7-GPJROM) |
| | SSN - Add Ping Job (D7-AddPingJobOBMsg) | SSN - Get Ping Results (D7-GetPingResultsOBMsg) |

### Silver Spring Networks MultiSpeak Outbound Message Types

This table lists the outbound message types used with the Silver Spring Networks adapter for MultiSpeak commands, along with the business objects used with each.

| Command | Outbound Message Type / Business Object |
|---|---|
| Device Commissioning | SSN - MultiSpeak Notify Meter Add (D7-MSNMA) |
| | SSN - MultiSpeak Meter Add Notification (D7-MSPMeterAddNotifyOBMsg) |
| Device Decommissioning | SSN - MultiSpeak Notify Meter Remove (D7-MSNMR) |
| | SSN - MultiSpeak Meter Remove Notification (D7-MSPMeterRemoveNotifyOBMsg) |
| Remote Connect | SSN - MSP Initiate Con/Discon Meter Number (D7-MSICDBMN) |
| | SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg) |
| Remote Disconnect | SSN - MSP Initiate Con/Discon Meter Number (D7-MSICDBMN) |
| | SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg) |
| On-Demand Read (MultiSpeak) | SSN - MSP Initiate Meter Read by Meter Id (D7-MSMRBMI) |
| | SSN - MSP Initiate Meter Read By Meter ID (D7-MSPInitMtrRdMtrIdOBMsg) |
| Device Status Check | SSN - MultiSpeak Initiate Outage Detection (D7-MSIOD) |
| | SSN - MultiSpeak Initiate Outage Detection (D7-MSPInitOutageDetectionOBMsg) |

## Creating Message Senders


Copyright © 2026, Oracle and/or its affiliates.

You must create message senders for each outbound communication that will be sent to the head end system as part of the command communication process.

> **Note**
>
> When creating message senders as part of the command communication process used with Smart Grid Gateway adapters, the **HTTP URL** must point to the appropriate server and head end system version. If an implementation has more than one version of a given head end system, each can have its own server and URL.

> **Note**
>
> When creating message senders as part of the command communication process used with native implementations, the **Message Class** MUST be set to **SOAPSNDR** (Sender for real-time HTTP / SOAP messages).

See Message Senders in the *Administrative User Guide* for more information about creating message senders.

## Creating an External System

You must create an external system that represents the head end system and defines a set of outbound message types that can be sent to the head end system.

### External System Outbound Message Types

You must configure an outbound message type on the external system for each type of message you will send to the head end system. An external system's outbound message types are defined by the following:

- **Outbound Message Type**: The outbound message type created for the message (see Creating Outbound Message Types)
- **Processing Method**: The method by which the message will be sent. This should be set to "Real-time" Smart Grid Gateway outbound communications.
- **Message Sender**: The message sender created for the message (see Creating Message Senders)
- **Message XSL / Response XSL**: XSL files used to transform your message and the response to the message into the appropriate format used by the head end system (Message XSL) and Smart Grid Gateway (Response XSL). Each Smart Grid Gateway adapter includes XSL files for each type of outbound communication.

See External Systems in the *Administrative User Guide* for more information about creating external systems.

### Itron OpenWay XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Itron OpenWay adapter.


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| Outbound Message for Itron Commission (D8-ItronCommissionOutboundMsg) | D8-CommissionRequest.xsl | D8-CommissionResponse.xsl |
| Outbound Message for Itron Decommission (D8-ItronDecommissionOutbndMsg) | D8-DecommissionRequest.xsl | D8-DecommissionResponse.xsl |
| Outbound Message for Itron connect command (D8-ItronRemoteConnectOutbndMsg) | D8-ReconnectRequest.xsl | D8-ReconnectResponse.xsl |
| Outbound Message for Itron connect result command (D8-ItronRemoteCntRsltOutMsg) | D8-GetReconnectRsRequest.xsl | D8-GetReconnectRsResponse.xsl |
| Outbound Message for Itron disconnect command (D8-ItronRemoteDisconnectOutMsg) | D8-DisconnectRequest.xsl | D8-DisconnectResponse.xsl |
| Outbound Message for Itron disconnect result command (D8-ItronRemoteDcntRsltOutMsg) | D8-GetDisconnectRsRequest.xsl | D8-GetDisconnectRsResponse.xsl |
| Outbound Message for Itron Contingency Read command (D8-ItronOnDemandReadOutbndMsg) | D8-ContRdByEndptsRequest.xsl | D8-ContRdByEndptsResponse.xsl |
| O/B Msg for Itron Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg) | D8-GetContRdByEndptsRsReq.xsl | D8-GetContRdByEndptsRsResp.xsl |
| Outbound Message for Itron Schedule Read command (D8-ItronInterrogateByGrpOBMsg) | D8-InterrogateByGroupReq.xsl | D8-InterrogateByGroupResp.xsl |
| Outbound Message for Itron Schedule Read Result command (D8-ItronInterrogateByGrpRsltOB) | D8-GetIntByGroupRsRequest.xsl | D8-GetIntByGroupRsResponse.xsl |
| Outbound Message for Itron Device status check (D8-ItronDvcStatusChkOutbndMsg) | D8-PingByEndpointsRequest.xsl | D8-PingByEndpointsResponse.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| O/B Msg for Itron DeviceStatusCheck Result (D8-ItronDvcStatusChkRstlOBMsg) | D8-GetPingByEndptsRsReq.xsl | D8-GetPingByEndptsRsResp.xsl |
| Outbound Message for Itron Multi device status check (D8-ItronDvcStChkMtDvcOutbndMsg) | D8-PingByEndpointsRequest.xsl | D8-PingByEndpointsResponse.xsl |
| O/B Message for Itron MultiDeviceStatusCheck Result (D8-ItronDvcStChkMtDvcRsltOBMsg) | D8-GetPingByEndptsRsReq.xsl | D8-GetPingByEndptsRsResp.xsl |
| Outbound Message for Itron Detect Load Side Voltage (D8-ItronDetectLoadSideVolOBMsg) | D8-DetLSVByMeterRequest.xsl | D8-DetLSVByMeterResponse.xsl |
| Outbound Message for LoadCheckResult Cmd (D8-ItronDetLSVoltMtrRsltOutMsg) | D8-GetDetLSVByMeterRsReq.xsl | D8-GetDetLSVByMeterRsResp.xsl |
| Outbound Message for Itron Read Disconnect State D8-ItronReadDisconStateOutMsg | D8-RdDiscStByMtrsRequest.xsl | D8-RdDiscStByMtrsResponse.xsl |
| Outbound Message for ReadDisconStateMeter Cmd D8-ItronRdDiscStByMtRsltOutMsg | D8-GetRdDiscStByMtrsRsReq.xsl | D8-GetRdDiscStByMtrsRsResp.xsl |

### Landis+Gyr XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Landis+Gyr adapter.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| LG - Add Meter To Inventory OB Message D3-AddMeterToInventoryOBMsg | D3-AddMtrToInvRequest.xsl | D3-AddMtrToInvResponse.xsl |
| LG - Meter Remove Notification OB Message D3-MeterRemoveNotifOBMsg | D3-MtrRemoveNotfRequest.xsl | D3-MtrRemoveNotfResponse.xsl |
| LG - Meter Retire Notification OB Message D3-MeterRetireNotifOBMsg | D3-MtrRetireNotfRequest.xsl | D3-MtrRetireNotfResponse.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| LG -Meter Exchange Notification OB Message<br>D3-MeterExchNotificatnOBMsg | D3-MtrExcNotifRequest.xsl | D3-MtrExcNotifResponse.xsl |
| L+G Initiate Connect Disconnect OB Message<br>D3-InitConnectDisconnectOBMsg | D3-InitiateConnDisconReq.xsl | D3-InitiateConnDisconResp.xsl |
| LG - Schedule Demand Reset OB Message<br>D3-ScheduleDemandResetOBMsg | D3-ScheduleDemandRstReq.xsl | D3-ScheduleDemandRstResp.xsl |
| L+G MultiSpeak InitiateMeterReadByMeterNumber OB Message<br>D3-InitMtrReadByMtrNumOBMsg | D3-InitMtrRdMtrNumRequest.xsl | D3-InitMtrRdMtrNumResponse.xsl |
| CIM AMIRequest GetMeterReads OB Message<br>D3-CIMAMIRequestGetMtrRdOBMsg | D3-CIMGetMtrReadsRequest.xsl | D3-CIMGetMtrReadsResponse.xsl |
| L+G CIM 2.0 - GetMeterReads OB Message<br>D3-CIMGetMtrRdOBMsg | D3-CIM2GetMtrReadsRequest.xsl | D3-CIM2GetMtrReadsResponse.xsl |

### Sensus XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Sensus adapter.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| Sensus - Meter Add Notification OB Message<br>D6-MeterAddNotificationOBMsg | D6-MtrAddRemoveNotifReq.xsl | D6-MeterAddNotifResponse.xsl |
| Sensus -Meter Remove Notification OB Msg<br>(D6-MeterRemoveNotificatnOBMsg) | D6-MtrAddRemoveNotifReq.xsl | D6-MeterAddNotifResponse.xsl |
| Sensus Initiate Connect Discon OB Message<br>(D6-InitConnectDisconnectOBMsg) | D6-InitiateConnDisconReq.xsl | D6-InitiateConnDisconResp.xsl |
| Sensus Initiate Connect Discon OB Message<br>(D6-InitConnectDisconnectOBMsg) | D6-InitiateConnDisconReq.xsl | D6-InitiateConnDisconResp.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| Sensus MultiSpeak InitiateMeterReadByMeterNumber OB Message (D6-InitMtrRdMtrIdOBMsg) | D6-InitMtrRdMtrIdRequest.xsl | D6-InitMtrRdMtrIdResponse.xsl |
| Sensus InitiateOutageDetection OB Message (D6-InitOutageDetectionOBMsg) | D6-InitOutDetectRequest.xsl | D6-InitOutDetectResponse.xsl |

### Silver Spring Networks XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Silver Spring Networks adapter.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| SSN - Find Device Outbound Message D7-FindDeviceOutboundMsg | D7-FindDeviceRequest.xsl | D7-FindDeviceResponse.xsl |
| SSN - Get Job Status Outbound Message D7-GetJobStatusOutboundMsgType | D7-GetJobStatusRequest.xsl | D7-GetJobStatusResponse.xsl |
| SSN - Replace Location (D7-ReplaceLocationDR) | D7-ReplaceLocationRequest.xsl | D7-ReplaceLocationResponse.xsl |
| SSN - Replace Device At Location (Decomm) (D7-ReplDeviceAtLocForDecommDR) | D7-RepDvcAtLocRequest.xsl | D7-RepDvcAtLocResponse.xsl |
| SSN - Connect or Disconnect (D7-ConnectDisconnectDR) | D7-AddRemProvJobRequest.xsl | D7-AddRemProvJobResponse.xsl |
| SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR) | D7-GetRemProvRsByJobIDReq.xsl | D7-GetRemProvRsByJobIDResp.xsl |
| SSN - Add Meter Read Job Scalar OB Msg D7-AddMeterReadJobScalarOBMsg | D7-AddMtrRdJobScRequest.xsl | D7-AddMtrRdJobScResponse.xsl |
| SSN - GetMeterReadResults(Scalar) By JobId D7-GetSclrMtrRdRstByJobIdOBMsg | D7-GetMtrRdRsByJobIDReq.xsl | D7-GetMtrRdRsByJobIDResp.xsl |
| SSN - AddMeterReadJob Interval OB Msg D7-GetIntMtrRdRsltByJobIdOBMsg | D7-AddMtrRdJobIntRequest.xsl | D7-AddMtrRdJobIntResponse.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|
| SSN - GetMeterReadResults(Interval) By JobId<br>D7-GetIMtrRdRsltJobIdOBMsgTyp | D7-GetMtrRdRsByJobIDReq.xsl | D7-GetMtrRdRsByJobIDResp.xsl |
| SSN - Add Ping Job<br>(D7-AddPingJobDR) | D7-AddPingJobRequest.xsl | D7-AddPingJobResponse.xsl |
| SSN - Ping Results<br>(D7-PingResultsDR) | D7-GetMtrRdRsByJobIDReq.xsl | D7-GetMtrRdRsByJobIDResp.xsl |

### Silver Spring NetworksMultiSpeak XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Silver Spring Networks adapter for MultiSpeak commands.

| Command | Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|---|
| Device Commissioning | SSN - MultiSpeak Meter Add Notification<br>(D7-MSPMeterAddNotifyOBMsg) | D7-MSMtrAddRemoveNotifReq.xsl | D7-MSMeterAddNotifResponse.xsl |
| Device Decommissioning | SSN - MultiSpeak Meter Remove Notification<br>(D7-MSPMeterRemoveNotifyOBMsg) | D7-MSMtrAddRemoveNotifReq.xsl | D7-MSMtrRemoveNotfResponse.xsl |
| Remote Connect | SSN - MSP Initiate Connect Disconnect<br>(D7-MSPInitConDisconOBMsg) | D7-MSInitiateConnDisconReq.xsl | D7-MSInitiateConnDisconRes.xsl |
| Remote Disconnect | SSN - MSP Initiate Connect Disconnect<br>D7-MSPInitConDisconOBMsg | D7-MSInitiateConnDisconReq.xsl | D7-MSInitiateConnDisconRes.xsl |
| On-Demand Read (MultiSpeak) | SSN - MSP Initiate Meter Read By Meter ID<br>D7-MSPInitMtrRdMtrIdOBMsg | D7-MSInitMtrRdMtrIdRequest.xsl | D7-MSInitMtrRdMtrIdRequest.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|---|
| Device Status Check | SSN - MultiSpeak Initiate Outage Detection<br>D7-MSPInitOutageDetectionOBMsg | D7-MSInitOutDetectRequest.xsl | D7-MSInitOutDetectResponse.xsl |

## Configuring Processing Methods

In order to send smart meter commands to a head end system, appropriate processing methods must be configured on the head end system record. The tables below list the details appropriate to configuring processing methods for each command.

This table includes the following:

- **Command**: The command you wish to configure. When a command is initiated for a device, the system creates a command activity.
- **Processing Role**: The Processing Role appropriate for the command to be configured.
- **Business Object**: The outbound communication business object to be created by the command activity. Note that most commands have separate business objects for the initial command request and the result request.
- **Outbound Message Business Object** The business object that should be used when creating outbound message types for each command. Note that most commands use separate business objects for the initial outbound message type and the result outbound message type.

### Itron OpenWay Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Itron OpenWay head end system.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Device Commissioning | Device Registration | Itron - Add Meter Definition (D8-AddMeterDefinitionsDR) | Outbound Message for Itron Commission<br>D8-ItronCommissionOutboundMsg |
| Device Decommissioning | Device Deregistration | Itron - Deregister Meter (D8-DeregisterMeterDR) | Outbound Message for Itron Decommission<br>D8-ItronDecommissionOutbndMsg |
| Remote Connect | Remote Connect | Itron - Reconnect Meter (Remote Connect)<br>D8-ReconnectMeterDR) | Outbound Message for Itron connect command<br>D8-ItronRemoteConnectOutbndMsg |
| | Get Remote Connect Result | Itron - Reconnect Meter Result (D8-ReconnectMeterResult) | Outbound Message for Itron connect result command<br>(D8-ItronRemoteCntRsltOutMsg) |
| Remote Disconnect | Remote Disconnect | Itron - Disconnect Meter (D8-DisconnectMeterDR) | Outbound Message for Itron disconnect command<br>D8-ItronRemoteDisconnectOutMsg |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| | Get Remote Disconnect Result | Itron - Disconnect Meter Result (D8-DisconnectMeterResult) | Outbound Message for Itron disconnect result command (D8-ItronRemoteDcntRsltOutMsg) |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | Itron - Contingency Read (Scalar) (D8-ReadScalarDR) | Outbound Message for Itron Contingency Read command D8-ItronOnDemandReadOutbndMsg |
| | Get On-Demand Read Result | Itron - On Demand Read Result (D8-ReadOnDemandReadResultDR) | O/B Msg for Itron Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg) |
| On-Demand Read (Interval) | On-Demand Read (Interval) | Itron - Contingency Read (Interval) (D8-ReadIntervalDR) | Outbound Message for Itron Contingency Read command (D8-ItronOnDemandReadOutbndMsg) |
| | Get On-Demand Read Result | Itron - On Demand Read Result (D8-ReadOnDemandReadResultDR) | O/B Msg for Itron Contingency Read Result (D8-ItronOnDemandReadRsltOBMsg) |
| Scheduled Read (Scalar) | Send Schedule Read Request | Itron - Interrogate by Group (Scalar) (D8-InterrogateByGroupDR) | Outbound Message for Itron Schedule Read command (D8-ItronInterrogateByGrpOBMsg) |
| | Get Schedule Read Result | Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR) | Outbound Message for Itron Schedule Read Result command (D8-ItronInterrogateByGrpRsltOB) |
| Scheduled Read (Interval) | Send Schedule Read Request | Itron - Interrogate by Group (Interval) (D8-InterrogateByGroupDR) | Outbound Message for Itron Schedule Read command (D8-ItronInterrogateByGrpOBMsg) |
| | Get Schedule Read Result | Itron - Interrogate by Group Results (D8-InterrogateByGroupResultDR) | Outbound Message for Itron Schedule Read Result command (D8-ItronInterrogateByGrpRsltOB) |
| Device Status Check | Device Status Check | Itron - Ping by Endpoints (Status Check) (D8-PingByEndpointsDR) | Outbound Message for Itron Device status check (D8-ItronDvcStatusChkOutbndMsg) |
| | Get Device Status Check Result | Itron - Ping By Endpoints Result (D8-PingByEndpointsResultDR) | O/B Msg for Itron DeviceStatusCheck Result (D8-ItronDvcStatusChkRstlOBMsg) |
| Multi-Device Status Check | Multi-Device Status Check | Itron - Ping By Endpoints Multi-Device (D8-PingByEndpointsMultiDvcDR) | Outbound Message for Itron Multi device status check (D8-ItronDvcStChkMtDvcOutbndMsg) |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| | Get Multi-Device Status Check Result | Itron – Ping By Endpoints Multi Dvc Result (D8-PingByEndpointsMDResultDR) | O/B Message for Itron MultiDeviceStatusCheck Result (D8-ItronDvcStChkMtDvcRsltOBMsg) |
| Load Check | Load Check | Itron - Detect Load Side Voltage (D8-DetectLoadSideVoltgeByMtrDR) | Outbound Message for Itron Detect Load Side Voltage (D8-ItronDetectLoadSideVolOBMsg) |
| | Get Load Check Results | Itron – Detect Load Side Voltage Result (D8-DetLoadSideVoltageMtrRsltDR) | Outbound Message for LoadCheckResult Cmd (D8-ItronDetLSVoltMtrRsltOutMsg) |

### Landis+Gyr Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Landis+Gyr head end system.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Device Commissioning | Device Registration | L+G Add Meter to Inventory(MultiSpeak) (D3-AddMeterToInventryMultiSpDR) | LG - Add Meter To Inventory OB Message D3-AddMeterToInventoryOBMsg |
| Device Decommissioning | Device Removal | L+G Meter Remove Notification (MultiSpeak) (D3-MtrRmvNotifMultiSpeakDR) | LG - Meter Remove Notification OB Message (D3-MeterRemoveNotifOBMsg) |
| Meter Retire | Device Deregistration | L+G Meter Retire Notification (D3-MeterRetireNotificationDR) | LG - Meter Retire Notification OB Message (D3-MeterRetireNotifOBMsg) |
| Meter Exchange | Device Installation | L+G Meter Exchange Notification (D3-MeterExNotificatnMultiSpDR) | LG -Meter Exchange Notification OB Message (D3-MeterExchNotificatnOBMsg) |
| Remote Connect | Remote Connect | L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR) | L+G Initiate Connect Disconnect OB Message (D3-InitConnectDisconnectOBMsg) |
| Remote Disconnect | Remote Disconnect | L+G Initiate Connect Disconnect DR (D3-InitiateConnectDisconnectDR) | L+G Initiate Connect Disconnect OB Message (D3-InitConnectDisconnectOBMsg) |
| Demand Reset | Demand Reset | L+G Schedule Demand Reset (Multispeak) (D3-ScheduleDemandResetDR) | LG - Schedule Demand Reset OB Message (D3-ScheduleDemandResetOBMsg) |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| On-Demand Read (MultiSpeak) | On-Demand Read (Scalar) | L+G MultiSpeak Initiate Meter Read By Meter Num Direct Route (D3-InitiateMRByMtrNbrDR) | L+G MultiSpeak InitiateMeterReadByMeterNumber OB Message (D3-InitMtrReadByMtrNumOBMsg) |
| On-Demand Read (Interval) [CIM] | On-Demand Read (Interval) | CIM Meter On Demand Read Direct Route (Interval) (D3-CIMGetLPDataDR) | CIM AMIRequest GetMeterReads OB Message (D3-CIMAMIRequestGetMtrRdOBMsg) |
| On-Demand Read (Scalar) [CIM] | On-Demand Read (Scalar) | CIM Meter On Demand Read Direct Route (Scalar) (D3-CIMMeterOnDemandReadDR) | CIM AMIRequest GetMeterReads OB Message (D3-CIMAMIRequestGetMtrRdOBMsg) |
| Device Status Check [CIM] | Device Status Check | CIM Ping Direct Route (D3-CIMPingDR) | CIM AMIRequest GetMeterReads OB Message (D3-CIMAMIRequestGetMtrRdOBMsg) |
| On-Demand Read (Interval) [CIM 2.0] | On-Demand Read (Interval) | L+G CIM 2.0 On-Demand Read DR (Interval) (D3-CIMGetMtrReadIntervalDR) | L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg) |
| On-Demand Read (Scalar) [CIM 2.0] | On-Demand Read (Scalar) | L+G CIM 2.0 - On-Demand Read DR (Scalar) (D3-CIMGetMtrReadScalarDR) | L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg) |
| Device Status Check [CIM 2.0] | Device Status Check | L+G CIM 2.0 - Ping DR (D3-CIM2PingDR) | L+G CIM 2.0 - GetMeterReads OB Message (D3-CIMGetMtrRdOBMsg) |

### Sensus Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Sensus head end system.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Device Commissioning | Device Registration | Sensus -Meter Add Notification Direct Route (D6-MeterAddNotificationDR) | Sensus - Meter Add Notification OB Message D6-MeterAddNotificationOBMsg |
| Device Decommissioning | Device Removal | Sensus -Meter Remove Notification Direct Route (D6-MeterRemoveNotificationDR) | Sensus -Meter Remove Notification OB Msg (D6-MeterRemoveNotificatnOBMsg) |
| Remote Connect | Remote Connect | Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR) | Sensus Initiate Connect Discon OB Message (D6-InitConnectDisconnectOBMsg) |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Remote Disconnect | Remote Disconnect | Sensus - Initiate Conn Discon - DR (D6-InitiateConnectDisconnectDR) | Sensus Initiate Connect Discon OB Message (D6-InitConnectDisconnectOBMsg) |
| On-Demand Read (MultiSpeak) | On-Demand Read (Scalar) | Sensus - Initiate Meter Read By Meter ID (D6-InitiateMeterByMeterIdDR) | Sensus MultiSpeak InitiateMeterReadByMeterNumber OB Message (D6-InitMtrRdMtrIdOBMsg) |
| Device Status Check | Device Status Check | Sensus Initiate Outage Detection DR (D6-InitiateOutageDetectionDR) | Sensus InitiateOutageDetection OB Message (D6-InitOutageDetectionOBMsg) |

### Silver Spring Networks Processing Methods

This table lists the details appropriate to configuring processing methods for each command supported for the Silver Spring Networks head end system.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Device Commissioning | Device Registration | SSN - Replace Location (D7-ReplaceLocationDR) | SSN - Replace Location (Comm) OB Message (D7-ReplaceLocationOBMsg) |
| Device Decommissioning | Device Removal | SSN - Replace Device At Location (Decomm) (D7-ReplDeviceAtLocForDecommDR) | SSN - Replace Device At Loc O/B Message: (D7-ReplaceDeviceAtLocOBMsg) |
| Remote Connect | Remote Connect | SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP) | SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg) |
| | Get Remote Connect Result | SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR) | SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg) |
| Remote Disconnect | Remote Disconnect | SSN - Add Remote Provisioning Job - Polling (D7-AddRemoteProvJobDRP) | SSN - Connect Disconnect OB Message (D7-ConnectDisconnectOBMsg) |
| | Get Remote Disconnect Result | SSN - Connect Disconnect Results (D7-ConnectDisconnectResultDR) | SSN - GetConnectDisconnectResult OB Message (D7-GetCntDiscntResultOBMsg) |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | SSN - Add Mtr Read Job (Scalar) - Polling (D7-AddMeterReadJobScalarDRP) | SSN - Add Meter Read Job Scalar OB Msg Type D7-AddMeterReadJobScalarOBMsg |
| | Get On-Demand Read Result (Scalar) | SSN - Meter Read Results (Scalar) (D7-MeterReadResultsScalar ) | SSN - GetMeterReadResults(Scalar) By JobId (D7-GetSMtrRdRsltJobIdOBMsgTyp ) |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| On-Demand Read (Interval) | On-Demand Read (Interval) | SSN - Add Mtr Read Job (Interval) - Polling (D7-AddMeterReadJobIntervalDRP) | SSN - AddMeterReadJob Interval OB Msg Type (D7-AddMtrReadJobIntervalOBMsg ) |
| | Get On-Demand Read Result (Interval) | SSN - Meter Read Results (Interval) (D7-MeterReadResultsInterval ) | SSN - GetMeterReadResults(Interval) By JobId (D7-GetIMtrRdRsltJobIdOBMsgTyp ) |
| Device Status Check | Device Status Check | SSN - Add Ping Job - Polling (D7-AddPingJobDRP) | SSN - Add Ping Job OB Message (D7-AddPingJobOBMsg) |
| | Get Device Status Check Result | SSN - Ping Results (D7-PingResultsDR) | SSN - Get Ping Results O/B Message (D7-GetPingResultsOBMsg) |
| Get Job Status | Get Job Status | SSN - Get Job Status for Polling (D7-GetJobStatusDRP) | SSN - Get Job Status Outbound Message (D7-GetJobStatusOutboundMsg) |

### Silver Spring Networks MultiSpeak Processing Methods

This table lists the details appropriate to configuring processing methods for each MultiSpeak command supported for the Silver Spring Networks head end system.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Device Commissioning | Device Registration | SSN - MultiSpeak Meter Add Notification DR (D7-MSPMeterAddNotificationDR) | SSN - MultiSpeak Meter Add Notification (D7-MSPMeterAddNotifyOBMsg) |
| Device Decommissioning | Device Removal | SSN - MSP Meter Remove Notification DR (D7-MSPMeterRemNotifyDR) | SSN - MultiSpeak Meter Remove Notification (D7-MSPMeterRemoveNotifyOBMsg) |
| Remote Connect | Remote Connect | SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR) | SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg) |
| Remote Disconnect | Remote Disconnect | SSN - MultiSpeak Initiate Conn Discon DR (D7-MSPInitConDisconDR) | SSN - MSP Initiate Connect Disconnect (D7-MSPInitConDisconOBMsg) |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| On-Demand Read (MultiSpeak) | On-Demand Read (Scalar) | SSN - MSP Initiate Read By Meter ID DR (D7-MSPInitMtrByMtrIdDR) | SSN - MSP Initiate Meter Read By Meter ID (D7-MSPInitMtrRdMtrIdOBMsg) |
| Device Status Check | Device Status Check | SSN - MSP Initiate Outage Detection DR (D7-MSPInitOutageDetectDR) | SSN - MultiSpeak Initiate Outage Detection (D7-MSPInitOutageDetectionOBMsg) |

# Receiving Inbound Communications

Receiving inbound communications from a head end system involves configuring the following:

- **Inbound Web Services**: These services are used to receive inbound communications from external systems. See Inbound Web Services in the *Administrative User Guide* for more information.
- **Master Configurations**: Defines configuration settings for individual adapters.
- Inbound Web Services

## Inbound Web Services

Inbound web services are used to receive inbound communications from external systems. The tables below list the inbound web services provided with each Smart Grid Gateway adapter.

### Itron OpenWay Inbound Web Services

The table below lists the inbound web service provided with the Itron OpenWay adapter.

| Name | Description |
|---|---|
| D8-RequestStatusChangedService | Itron RequestStatusChangedService: Use to receive “status changed” messages from the Itron OpenWay head end system.<br>This service creates instances of the “Itron – StatusChanged” (D8-StatusChanged) inbound communication business object for each message. |


Copyright © 2026, Oracle and/or its affiliates.

| Name | Description |
|---|---|
| D8-DataSubscriberService | Itron DataSubscriberService: Used to receive usage and device event payloads from the Itron OpenWay head end system.<br>This service invokes the “Itron – DataArrived Request” (D8_DATAARR) service script, which invokes the ItronDataArrivedStreamProcessor business component to process the usage or device event payload based on the configuration of the Itron Master Configuration, the SGG Payload Processing Configuration extendable lookup, and the File Storage extendable lookup. |
| D8-EventSubscriberService | Itron EventSubscriberService: Used to receive device event payloads from the Itron OpenWay head end system.<br>This service invokes the “Itron – EventsArrived Request” (D8_EVNTSARR) service script, which invokes the ItronEventsArrivedStreamProcessor business component to process the device event payload based on the configuration of the Itron Master Configuration, the SGG Payload Processing Configuration extendable lookup, and the File Storage extendable lookup. |
| D8-ExceptionSubscriberService | Itron ExceptionSubscriberService: Used to receive exceptions from the Itron OpenWay head end system.<br>This service invokes the “Itron – ExceptionsArrived Request” (D8_EXCPARR) service script, which invokes the ItronExceptionsArrivedStreamProcessor business component to process the exception based on the configuration of the Itron Master Configuration, the SGG Payload Processing Configuration extendable lookup, and the File Storage extendable lookup. |

### Landis+Gyr Inbound Web Services

The table below lists the inbound web services provided with the Landis+Gyr adapter.

| Name | Schema Name | Request / Response XSL |
|---|---|---|
| CD State Changed Notification (Callback)<br>(D3-CB_CD) | D3-ConnectDisconStateChgNtfDR | D3-ConDiscnStateChgNtfReq.xsl<br>D3-ConDiscnStateChgNtfResp.xsl |
| Reading Changed Notification (Callback)<br>(D3-CB_MR) | D3-ReadingChgNotificationDR | D3-ReadingChgNotifRequest.xsl<br>D3-ReadingChgNotifResponse.xsl |
| CIM Service Callback<br>(D3-CIMCallback) | D3-CIMAsyncReplyMeterReadingDR | D3-CIMMeterReadingRequest.xsl<br>D3-CIMMeterReadingResponse.xsl |
| Schedule Demand Response (Multispeak)<br>(D3-ScheduleDemandResetResponse) | D3-ReadingChgNotificationDR | D3-ReadingChgNotifRequest.xsl<br>D3-ReadingChgNotifResponse.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Name | Schema Name | Request / Response XSL |
|---|---|---|
| L+G CIM 2.0 - Meter Read Response Callback<br>(D3-CIM2ODRResponse) | D3CIM2MDIn | D3-CIMMtrReadingsRequest.xsl<br>D3-CIMMtrReadingsRequest.xsl |

### Sensus Inbound Web Services

The table below lists the inbound web services provided with the Sensus adapter.

| Name | Schema Name | Request / Response XSL |
|---|---|---|
| Reading Changed Notification (Callback)<br>(D6-CB_MR) | D6-ReadingChgNotificationDR | D6-ReadingChgNotifRequest.xsl<br>D6-ReadingChgNotifResponse.xsl |
| Sensus MultiSpeak CB_Server Interface<br>(D6-CB_Server) | D6-ConnectDisconStateChgNtfDR | D6-ConDiscnStateChgNtfReq.xsl<br>D6-ConDiscnStateChgNtfResp.xsl |
| | D6-ReadingChgNotificationDR | D6-ReadingChgNotifRequest.xsl<br>D6-ReadingChgNotifResponse.xsl |
| Sensus MultiSpeak OA_Server Interface<br>(D6-OA_Server) | D6-ODEventNotificationDR | D6-ODEventNotifRequest.xsl<br>D6-ODEventNotifResponse.xsl |
| Unsolicited Event Response<br>D6-UnsolicitedEventNotification | D6-UnsolicitedEvtNotification | |

### Silver Spring Networks MultiSpeak Inbound Web Services

The table below lists the inbound web services provided with the Silver Spring Networks adapter for use with MultiSpeak commands.

| Name | Operation | Schema Name | Request / Response XSL |
|---|---|---|---|
| SSN - MultiSpeak CB_Server Interface<br>(D7-MS_CB_Server) | CDStatesChangedNotification | SSN - MultiSpeak CD State Change Notification Inbound Request<br>(D7MSCDSCNIR) | D7-MSConDiscnStaChgNtfReq.xsl<br>D7-MSConDiscnStaChgNtfRes.xsl |
| | ReadingChangedNotification | SSN - MultiSpeak Reading Change Notification Inbound Request<br>(D7MSReChNtIR) | D7-MSReadingChgNotiRequest.xsl<br>D7-MSReadChgNotiResponse.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Name | Operation | Schema Name | Request / Response XSL |
|---|---|---|---|
| SSN - MultiSpeak OA_Server Interface (D7-MS_OA_Server) | ODEventNotification | SSN - MultiSpeak OD Event Notification Inbound Request (D7MSODEvNtIR) | D7-MSODEventNotifRequest.xsl D7-MSODEventNotifResponse.xsl |

# Adapter Development Kit - MultiSpeak Commands

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process smart meter commands in the MultiSpeak format.

These commands can be used to support smart meter commands for meters for unsupported head end systems that use the MultiSpeak 4.1 protocol.

- Adapter Development Kit MultiSpeak Command Flows
- Adapter Development Kit MultiSpeak Command Extendable Lookups
- Sending Adapter Development Kit MultiSpeak Command Outbound Communications
- Receiving Adapter Development Kit MultiSpeak Command Inbound Communications

## Adapter Development Kit MultiSpeak Command Flows

The table below lists the communication flows used with each Silver Spring Networks MultiSpeak command, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Commissioning (MultiSpeak) | ADK - MultiSpeak Meter Add Notification DR (DG-MSPMeterAddNotificationDR) | | Device Commissioning Completion Event |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Decommissioning (MultiSpeak) | ADK - MSP Meter Remove Notification DR (DG-MSPMeterRemNotifyDR) | | Device Decommissioning Completion Event |
| Remote Connect (MultiSpeak) | ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR) | ADK - MSP Conn/ Discon State Change DR (DG-MSPConDisconStaChgNtfDR) | Connect Device Completion Event |
| Remote Disconnect (MultiSpeak) | ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR) | ADK - MSP Conn/ Discon State Change DR (DG-MSPConDisconStaChgNtfDR) | Disconnect Device Completion Event |
| On-Demand Read (MultiSpeak) | ADK - MSP Initiate Read By Meter ID DR (DG-MSPInitMtrByMtrIdDR) | ADK - MSP Reading Changed Notification DR (DG-MSPReadingChgNotifyDR) | Create IMD Completion Event |
| Device Status Check (MultiSpeak) | ADK - MSP Initiate Outage Detection DR (DG-MSPInitOutageDetectDR) | ADK - MSP OD Event Notification DR (DG-MSPODEventNotificationDR) | |

### Unit of Measure Filtering for On-Demand Reads

Readings returned via On-Demand Read commands are filtered on the combination of Unit of Measure/Time of Use/Service Quantity Identifiers (UOM/TOU/SQI) configured for the device and measuring component. When filtering is enabled, only readings that match (configured UOM/TOU/SQI combination are processed. All others are filtered o (ut.

Filtering is enabled by default via the “Opt Out Of Filtering” parameter (set to "false" by default) on the MSP Create IMD Completion Event DR (D1-MSCCEDR) algorithm on the Create Completion Events status of the MultiSpeak Reading Changed Notification DR (D1-MSPReadingChgNotifyDR) business object.

The MSP Check if Device has any Filtering errors DR (D1-MSVADVFDR) algorithm on the Validate status of the D1-MSPReadingChgNotifyDR business object is used to check for errors related to filtering. The value of the “Opt Out Of Filtering” and "Xpath to External UOM" parameters on this algorithm must match the values of the same parameters on the D1-MSCCEDR algorithm. The “Opt Out Of Filtering” parameters should both be set to either "false" (the default) or "true" (if filtering is to be disabled).

## Adapter Development Kit MultiSpeak Command Extendable Lookups


Copyright © 2026, Oracle and/or its affiliates.

Processing of MultiSpeak commands requires configuration of values for a number of extendable lookups. This topic lists the extendable lookups used with processing MultiSpeak commands using the Adapter Development Kit.

- MultiSpeak Connect/Disconnect Reason Code (D1-MSPCDReasonCodeLookup)
- MultiSpeak Header Build Descriptor (D1-MSPBuildDescriptorLookup)
- MultiSpeak Header Coordinate System Units (D1-MSPCSUnitsLookup)
- MultiSpeak Header Message Context (D1-MSPMessageContextLookup)
- MultiSpeak Load Action Code (D1-MSPLoadActionCodeLookup)
- MultiSpeak Service Type (D1-MSPServiceTypeLookup)
- MultiSpeak Service Type MappingD1-MSPServiceTypeMappingLookup)
- MultiSpeak Time Units (D1-MSPTimeUnits)
- MultiSpeak Verb Action Code (D1-MSPActionCodeLookup)

## Sending Adapter Development Kit MultiSpeak Command Outbound Communications

This topic describes the configuration required to process MultiSpeak outbound communications.

### Adapter Development Kit MultiSpeak Outbound Message Types

Outbound message types must be configured for each command you plan to use in your implementation. The base package includes a set of outbound message types that can be used.

This table lists the outbound message types used with Adapter Development Kit MultiSpeak commands, along with the business objects used with each.

| Command | Outbound Message Type / Business Object |
|---|---|
| Device Commissioning | ADK - MultiSpeak Notify Meter Add (DG-MSPMANOMT)<br>ADK - MultiSpeak Meter Add Notification (DG-MSPMeterAddNotifyOBMsg) |
| Device Decommissioning | ADK - MultiSpeak Notify Meter Remove (DG-MSPMRNOMT)<br>ADK - MultiSpeak Meter Remove Notification (DG-MSPMeterRemoveNotifyOBMsg) |
| Remote Connect | ADK - MultiSpeak Initiate Connect Disconnect by Meter Number (DG-MSPICBOMT)<br>ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg) |
| Remote Disconnect | ADK - MultiSpeak Initiate Connect Disconnect by Meter Number (DG-MSPICBOMT)<br>ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg) |
| On-Demand Read (MultiSpeak) | ADK- MultiSpeak Initiate Meter Read by Meter Id (DG-MSPMRBMIO)<br>ADK - MSP Initiate Meter Read By Meter ID (DG-MSPInitMtrRdMtrIdOBMsg) |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Message Type / Business Object |
|---|---|
| Device Status Check | ADK - MultiSpeak Initiate Outage Detection (DG-MSPIODOMT)<br>ADK - MultiSpeak Initiate Outage Detection (DG-MSPInitOutageDetectionOBMsg) |

### Adapter Development Kit MultiSpeak Message Senders

You must create message senders for each outbound communication that will be sent to the head end system as part of the command communication process.

Note: When creating message senders as part of the command communication process used with native implementations, the Message Class MUST be set to SOAPSNDR (Sender for real-time HTTP / SOAP messages).

See **Message Senders** in the *Administrative User Guide* for more information about creating message senders.

### Adapter Development Kit MultiSpeak External System

You must create an external system that represents the head end system and defines a set of outbound message types that can be sent to the head end system.

### Adapter Development Kit MultiSpeak XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided with the Adapter Development Kit for MultiSpeak commands.

| Command | Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|---|
| Device Commissioning | ADK - MultiSpeak Meter Add Notification (DG-MSPMeterAddNotifyOBMsg) | DG-MSMtrAddRemoveNotifReq.xsl | DG-MSMeterAddNotifResponse.xsl |
| Device Decommissioning | ADK - MultiSpeak Meter Remove Notification (DG-MSPMeterRemoveNotifyOBMsg) | DG-MSMtrAddRemoveNotifReq.xsl | DG-MSMtrRemoveNotfResponse.xsl |
| Remote Connect | ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg) | DG-MSInitiateConnDisconReq.xsl | DG-MSInitiateConDisconResp.xsl |
| Remote Disconnect | ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg) | DG-MSInitiateConnDisconReq.xsl | DG-MSInitiateConDisconResp.xsl |
| On-Demand Read (MultiSpeak) | ADK - MSP Initiate Meter Read By Meter ID (DG-MSPInitMtrRdMtrIdOBMsg) | DG-MSInitMtrRdMtrIdRequest.xsl | DG-MSIniMtrRdMtrIdResponse.xsl |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|---|
| Device Status Check | ADK - MultiSpeak Initiate Outage Detection (DG-MSPInitOutageDetectionOBMsg) | DG-MSInitOutDetectRequest.xsl | DG-MSInitOutDetectResponse.xsl |

See **External Systems** in the *Administrative User Guide* for more information about creating exteral systems.

### Adapter Development Kit MultiSpeak Processing Methods

In order to send smart meter commands to a head end system, appropriate processing methods must be configured on the head end system record.

This table lists the details appropriate to configuring processing methods for each MultiSpeak command used with the Adapter Development Kit.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Device Commissioning | Device Registration | ADK - MultiSpeak Meter Add Notification DR (DG-MSPMeterAddNotificationDR) | ADK - MultiSpeak Meter Add Notification (DG-MSPMeterAddNotifyOBMsg) |
| Device Decommissioning | Device Removal | ADK - MSP Meter Remove Notification DR (DG-MSPMeterRemNotifyDR) | ADK - MultiSpeak Meter Remove Notification (DG-MSPMeterRemoveNotifyOBMsg) |
| Remote Connect | Remote Connect | ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR) | ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg) |
| Remote Disconnect | Remote Disconnect | ADK - MultiSpeak Initiate Conn Discon DR (DG-MSPInitConDisconDR) | ADK - MSP Initiate Connect Disconnect (DG-MSPInitConDisconOBMsg) |
| On-Demand Read (MultiSpeak) | On-Demand Read (Scalar) | ADK - MSP Initiate Read By Meter ID DR (DG-MSPInitMtrByMtrIdDR) | ADK - MSP Initiate Meter Read By Meter ID (DG-MSPInitMtrRdMtrIdOBMsg) |
| Device Status Check | Device Status Check | ADK - MSP Initiate Outage Detection DR (DG-MSPInitOutageDetectDR) | ADK - MultiSpeak Initiate Outage Detection (DG-MSPInitOutageDetectionOBMsg) |


Copyright © 2026, Oracle and/or its affiliates.

See **Configuring Processing Methods** for more information.

## Receiving Adapter Development Kit MultiSpeak Command Inbound Communications

Inbound web services are used to receive inbound communications from external systems.

The table below lists the inbound web services provided with the Adapter Development Kit for use with MultiSpeak commands.

| Name | Operation | Schema Name | Request / Response XSL |
|---|---|---|---|
| ADK MultiSpeak CB_Server Interface (DG-CB_Server) | CDStatesChangedNotification | MultiSpeak Connect Disconnect State Change Notification Inbound Request (DGMSCDSCNIR) | DG-MSConDiscStateChgNtfReq.xsl<br>DG-MSConDisStateChgNtfResp.xsl |
| | ReadingChangedNotification | MultiSpeak Reading Change Notification Inbound Request (DGMSReChNtIR) | DG-MSReadingChgNotRequest.xsl<br>DG-MSReadingChgNotResponse.xsl |
| ADK MultiSpeak OA_Server Interface (DG-OA_Server) | ODEventNotification | MultiSpeak OD Event Notification Inbound Request (DGMSODEvNtIR) | DG-MSODEventNotifRequest.xsl<br>DG-MSODEventNotifResponse.xsl |

See **Inbound Web Services** for more information.

# Adapter Development Kit - Common Information Model Commands

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to process smart meter commands in the Common Information Model (CIM) format.

- CIM Commands Introduction and Overview
- Adapter Development Kit CIM Command Flows
- CIM Command Extendable Lookups
- Sending CIM Command Outbound Communications
- Receiving CIM Command Inbound Communications

## CIM Commands Introduction and Overview

The Adapter Development Kit supports smart meter commands in the Common Information Model (CIM) format. The CIM formats and standards supported by the Adapter Development Kit are based on the International Standard (IEC 61968-9, Edition 2.0), available **here**. Reference to specific data elements from the CIM command messages used in this section are based on this document. Refer to this document for detailed information about the CIM commands.


Copyright © 2026, Oracle and/or its affiliates.

The table below lists the communication flows used with Common Information Model (CIM) commands, including:

- **Command**: The specific smart meter command
- **Outbound Communication**: The outbound communication business object created by the command. This sends the command request to the head end system.
- **Inbound Communication**: The inbound communication created by the response from the head end system
- **Completion Event**: The completion event(s) triggered by the command, if any. Completion events are used to update data in the system as a result of a smart meter command. For example, if a command changes the status of a device, the completion event is responsible for making that change.

| Command | Outbound Communication | Inbound Communication | Completion Event |
|---|---|---|---|
| Device Status Check | ADK CIM Ping Direct Route (DG-CIMPingDR) | ADK CIM Ping Response Direct Route (DG-CIMPingResponseDR) | |
| On-Demand Read (Interval) [CIM] | ADK CIM On-Demand Read Direct Route (Interval) (DG-CIMGetMtrReadIntervalDR) | ADK CIM On-Demand Read Response Direct Route (DG-CIMMtrReadResponseDR) | Create IMD Completion Event |
| On-Demand Read (Scalar) [CIM] | ADK CIM On-Demand Read Direct Route (Scalar) (DG-CIMGetMtrReadScalarDR) | ADK CIM On-Demand Read Response Direct Route (DG-CIMMtrReadResponseDR) | Create IMD Completion Event |

### Configuring Command Types

The following communication business objects (from the table above) all have corresponding communication types that must be configured before communications based on those business objects can be created. The table below lists the communication type business objects used with each.

| Communication Business Object | Communication Type Business Object |
|---|---|
| ADK CIM Ping Direct Route (DG-CIMPingDR) | ADK - CIM Ping DR Type (DG-CIMPingDRType) |
| ADK CIM Ping Response Direct Route (DG-CIMPingResponseDR) | ADK - CIM Ping Async Response DR Type (DG-CIMPingResponseDRType) |
| ADK CIM On-Demand Read Direct Route (Interval) (DG-CIMGetMtrReadIntervalDR) | ADK - CIM On-Demand Read (Interval) DR Type (DG-CIMGetMtrReadIntervalType) |
| ADK CIM On-Demand Read Direct Route (Scalar) (DG-CIMGetMtrReadScalarDR) | ADK - CIM On-Demand Read (Scalar) DR Type (DG-CIMGetMtrReadScalarType) |
| ADK CIM On-Demand Read Response Direct Route (DG-CIMMtrReadResponseDR) | ADK - CIM AsyncReply MeterReads DR Type (DG-CIMMtrReadResponseDRType) |


Copyright © 2026, Oracle and/or its affiliates.

Refer to **Communication Types** for more information about communication types.

## CIM Command Extendable Lookups

Processing of CIM commands requires configuration of values for a number of extendable lookups. This topic outlines the extendable lookups used with processing CIM commands.

### Payload Processing Extendable Lookups

Processing of measurement data returned via on-demand read commands requires configuration of values for the following extendable lookups:

- **Unit of Measure Mapping**: Unit of measure mapping is based on the ReadingType string in the CIM payload and uses the ADK - UOM Code to Standard UOM Mapping (DG-HeadendUOMLookup) extendable lookup.
- **Interval Size Mapping**: Interval size (expressed as seconds per interval or SPI) mapping is based on the measuringPeriod value extracted from the ReadingType string and uses the ADK CIM Measuring Period Lookup (DG-CIMMeasuringPeriodLookup) extendable lookup.
- **Condition Code Mapping**: Condition code mapping is based on the ReadingQualityType string in the CIM payload (if present) and uses the the ADK - Interval Sts Code to Cond Map (DG-IntStsCodeToCondMapLookup) extendable lookup.

See **CIM Payload Processing Data Mapping and Extendable Lookups** for more information about these extendable lookups.

### ADK CIM Header Verb Lookup

The ADK CIM Header Verb Lookup (DG-CIMHeaderVerbLookup) extendable lookup defines values for message type verbs used in CIM messages. Refer to Annex A in the **International Standard document** (LINK) for detailed information about verbs. This extendable lookup is pre-populated with values from the International Standard document.

### ADK CIM Reply Code Lookup

The ADK CIM Reply Code Lookup (DG-CIMReplyCodeLookup) extendable lookup contains the enumeration of the values in the Reply / Error / Code element in replies from the head end system. The values are defined in Annex B in the **International Standard document** (LINK). Values comprise two numbers separated by a dot. Example: ReplyCode = <ReplyCode Category> “.” <ReplyCode Index>

This extendable lookup is pre-populated with values from **Table B.10 – Common enumerations** in the International Standard document.

### ADK CIM Reply Result Lookup

The ADK CIM Reply Result Lookup (DG-CIMResponseReplyRsltLookup) extendable lookup contains the enumeration of the values for the Reply.Result value returned from the head end system. This extendable lookup is pre-populated with values from the International Standard document (Refer to Annex B).

### ADK CIM Data Source Extendable Lookup

The ADK CIM Data Source Extendable Lookup (DG-DataSourceLookup) extendable lookup is used to map descriptions to data sources defined in the head end system. Each value defined for the CIM Data Source extendable lookup should include the following:

- **Data Source**: The CIM code for the data source


Copyright © 2026, Oracle and/or its affiliates.

- **Description**: A description of the data source
- **Status**: The status of the lookup value (can be Active or Inactive)

This extendable lookup is pre-populated with two values : AMI and Database.

## Sending CIM Command Outbound Communications

This topic describes the configuration required to process (CIM) outbound communications.

### Adapter Development Kit CIM Outbound Message Types

Outbound message types must be configured for each command you plan to use in your implementation. The base package includes a set of outbound message types that can be used.

This table lists the outbound message types used with Adapter Development Kit CIM commands, along with the business objects used with each.

| Command | Outbound Message Type / Business Object |
|---|---|
| Device Status Check | ADK - CIM Get MeterReadings (DG-CIMGMR) / ADK CIM Get Meter Reads (DG-CIMGetMtrRdOBMsg) |
| On-Demand Read (Interval) | ADK - CIM Get MeterReadings (DG-CIMGMR) / ADK - CIM Get Meter Reads (DG-CIMGetMtrRdOBMsg) |
| On-Demand Read (Scalar) | ADK - CIM Get MeterReadings (DG-CIMGMR) / ADK - CIM Get Meter Reads (DG-CIMGetMtrRdOBMsg) |

See **Outbound Messages** in the *Administrative User Guide* for more information about creating message senders.

### Adapter Development Kit CIM Message Senders

You must create message senders for each outbound communication that will be sent to the head end system as part of the command communication process.

Note: When creating message senders as part of the command communication process used with native implementations, the Message Class MUST be set to SOAPSNDR (Sender for real-time HTTP / SOAP messages).

See **Message Senders** in the *Administrative User Guide* for more information about creating message senders.

### Adapter Development Kit CIM External System

You must create an external system that represents the head end system and defines a set of outbound message types that can be sent to the head end system.

### Adapter Development Kit CIM XSL Files

The table below lists the XSL provided for use with outbound message types based on the outbound message business objects provided for use with Adapter Development Kit CIM commands.


Copyright © 2026, Oracle and/or its affiliates.

| Outbound Message Type | Outbound Message Business Object | Message XSL | Response XSL |
|---|---|---|---|
| ADK - CIM Get MeterReadings (DG-CIMGMR) | ADK - CIM Get Meter Reads(DG-CIMGetMtrRdOBMsg) | DG-CIMGetMtrReadsRequest.xsl | DG-CIMGetMtrReadsResponse.xsl |

### Adapter Development Kit CIM Processing Methods

In order to send smart meter commands to a head end system, appropriate processing methods must be configured on the head end system record.

This table lists the details appropriate to configuring processing methods for each command used with the Adapter Development Kit.

| Command | Processing Role | Business Object | Outbound Message Business Object |
|---|---|---|---|
| Device Status Check | Device Status Check | ADK CIM Ping Direct Route (DG-CIMPingDR) | ADK CIM GetMeterReads OB Message (DG-CIMGetMtrRdOBMsg) |
| On-Demand Read (Scalar) | On-Demand Read (Scalar) | ADK CIM On-Demand Read Direct Route (Scalar) (DG-CIMGetMtrReadScalarDR) | ADK CIM GetMeterReads OB Message (DG-CIMGetMtrRdOBMsg) |
| On-Demand Read (Interval) | On-Demand Read (Interval) | ADK CIM On-Demand Read Direct Route (Interval) (DG-CIMGetMtrReadIntervalDR) | ADK CIM GetMeterReads OB Message (DG-CIMGetMtrRdOBMsg) |

See **Configuring Processing Methods** for more information.

## Receiving CIM Command Inbound Communications

Inbound web services are used to receive inbound communications from external systems.

The table below lists the inbound web service provided for use with Adapter Development Kit CIM commands.

| Name | Description |
|---|---|
| DG-CIMPingResponse | This Inbound Web Service receives incoming asynchronous callback requests for Device Status Check. This web service accepts requests for the Response method defined in {http://iec.ch/TC57/2011/abstract}Generic.wsdl. |
| DG-CIMODRResponse | This Inbound Web Service receives incoming asynchronous callback requests for On-Demand Reads. This web service accepts requests for the Response method defined in {http://iec.ch/TC57/2011/abstract}Generic.wsdl. |

See **Inbound Web Services** for more information.


Copyright © 2026, Oracle and/or its affiliates.

# Adapter Development Kit - Creating Custom Commands

This section describes how the Oracle Utilities Smart Grid Gateway Adapter Development Kit can be configured to support custom smart meter commands, including those from non-supported head end systems.

This includes:

- An overview of the data and objects used by smart meter commands
- A description and example of synchronous commands
- A description and example of asynchronous commands


- Overview
- Synchronous Commands
- Asynchronous Commands


## Overview

This section provides an overview of how smart meter commands work and the types of objects used in executing commands when implementing Smart Grid Gateway adapters using native implementations and with Oracle Utilities cloud services.

As noted inDevice Communication Overview, command processing works slightly differently than it does in middleware implementations. That section provides an introduction to command processing in Smart Grid Gateway adapters, and the **Native Implementation Command Processing** section provides a step-by-step description of the various data and objects created as part of command processing.

The remainder of this section provides additional details regarding the three primary types of data and objects used with smart meter command processing:

- Command Activities
- Outbound Communications
- Inbound Communications


- Command Activities
- Outbound Communications
- Inbound Communications


## Command Activities

Smart meter commands are managed by activities. Activities orchestrate one or more communications and execute any completion events associated with the command.

### Key Algorithms

This section outlines some of the important algorithms executed as part of the command activity's lifecycle.

Some algorithms are directly shared by multiple business objects and multiple lifecycle states. In other cases, unique algorithms will perform the same role across business objects. The following section refers to the algorithms generically.



Copyright © 2026, Oracle and/or its affiliates.

| Lifecycle State | Algorithms |
| --- | --- |
| Validate | The Validate state is an opportunity to check whether the command should execute.<br>• Check Device Command Eligibility: Checks whether the command request is eligible for the device by calling the Determine Eligible Commands Business Service (D1-DetermineEligibleCommands). The command request is considered eligible if it exists in the eligible commands list returned by the business service.<br>• Validate Head End's Capability to Perform Activity: This algorithm checks to ensure that the head end system has the capability to support the command requested. It accomplishes this by invoking the Determine Service Providers and Methods Business Service.<br>• Check for Existing Active Request: This algorithm looks for non-Final Activities with the same Device and of the same type. This prevents duplicate requests from being sent to the head end system. |
| Validation Error | If the Activity fails validation, it transitions to the Validation Error state. From this state the activity can be retried and create To Do entriesto notify users of the error.<br>• Retry BO In Error: Reprocesses the BO by transitioning it to its original state.<br>• Create To Do Entry for BO in Error: This algorithm attempts to create a To Do entry using the To Do type and role specified by the algorithm parameters.<br>• Generic To Do Completion for BOs: This algorithm completes all To Do entries with Drill Keys = Current Business Object's prime keys except for To Do Types with Characteristic Type and Value as specified by algorithm parameters. This prevents the existence of multiple To Do entries for the same command and device. |
| Waiting For Effective Date | Prior to sending the request to the head end system, the activity enters this state.<br>• Wait Time Out - Transition to Rejection: This algorithm rejects the business object if the expiration date time has been reached.<br>• Wait for Effective Date: When the required date is reached, this algorithm transitions the business object to the next configured state.<br>• Send Response to External Requester: This algorithm sends an acknowledging received response outbound message to the requester if the business object is configured to do so. |
| Connection Ready | The Connection Ready state shows that the activity can be executed and creates the proper outbound communication business object. This state can be re-entered when there are multiple outbound communications that are required by the head end system's API.<br>• Create Outbound Communication: This algorithm creates an outbound communication instance if the value of the completion flag specified by an algorithm parameter is false.<br>• Create Results Outbound based on Completion Flag: This algorithm triggers a secondary communication to the head end system based on the value of an algorithm parameter. Typically, this indicates the successful completion of an initial outbound communication. |


Copyright © 2026, Oracle and/or its affiliates.

| Lifecycle State | Algorithms |
|---|---|
| Communication In Progress | This lifecycle state indicates that an outbound communication is in-progress.<br>• Wait Time Out - Transition to Exception: If the business object has been in the In Progress state too long, this algorithm transitions the business object to an exception state.<br>• Check Children Communications: If all of the outbound communication business object instances related to this activity have completed, the algorithm transitions the business object to the next state. |
| Communication Error | An error-handling lifecycle state to which the business object transitions when the communication encounters a problem. The algorithms are similar to those in the Validation Error state.<br>• Wait Time Out - Transition to Rejection: If the business object has been in the In Progress state too long, this algorithm transitions the business object to a rejection state.<br>• Retry BO In Error: This algorithm reprocesses the business object by transitioning it to its original state.<br>• Create To Do Entry for BO in Error: This algorithm attempts to create a To Do entry using the To Do type and role specified by the algorithm parameters.<br>• Generic To Do Completion for BOs: This algorithm completes all To Do entries with Drill Keys = Current Business Object's prime keys except for To Do Types with Characteristic Type and Value as specified by algorithm parameters. This prevents the existence of multiple To Do entries for the same command and device |
| Retry | The lifecycle state responsible for discarding existing outbound communications and transitioning the activity back to Connection Ready. |
| Execute Completion Events | This lifecycle state collates the information acquired by the communications and creates some result. For example, in a DeviceStatusCheck, the overall results are set.<br>• Execute Completion Events: Fires the completion events associated with the activity. |
| Completion Event Error | This is another error-handling state similar to the Validation Error and the Communication Error states. The algorithms are the same as in those states. |
| Completed | This is the final state representing a successful activity. Finishing tasks are performed and external requesters are notified. |
| Discarded | Similarly to the Completed state, the Discarded state finishes the activity and indicates an overall failure. External requesters are notified of the failure and other finishing tasks are performed. |

## Outbound Communications

Outbound Communications represent the business logic of sending a request and receiving a reply. Outbound communication business objects are based on the D1-COMMOUT maintenance object. The outbound communication is not the actual message, but creates the outbound message. Outbound communication business objects used with native implementations are suffixed with “-DR” for “direct route” to indicate the request go straight to the head end system.


Copyright © 2026, Oracle and/or its affiliates.

### Schema

This section provides details concerning outbound communication business object schemas.

Outbound communication schemas typically have a “sendDetail” element corresponding to the data being sent to the head end system, or more precisely, the XSL handler for the outbound message. The synchronous response is transformed by the XSL handler and placed into the “responseDetail” element.

### Key Algorithms

This section outlines some of the important algorithms executed as part of the outbound communication’s lifecycle.

Some algorithms are directly shared by multiple business objects and multiple lifecycle states. In other cases, unique algorithms will perform the same role across business objects. The following section refers to the algorithms generically.

| Lifecycle State | Algorithms |
|---|---|
| Validate | Similar to the sctivity, the Validate state houses algorithms that will execute prior to sending the message.<br>• Validate Communication Type: This algorithm ensures the communicationType field is populated on the message. |
| Validation Error | If the Validate state encounters an error, the business object transitions to this state. The algorithms are similar to those in the activity business object. |
| Awaiting Response | The Awaiting Response state sends an outbound message and awaits the response.<br>• Populate Send Detail: Each outbound communication has a specific algorithm to populate all the schema elements that are required to create the outbound message.<br>• Create Outbound Message: This algorithm creates a specific outbound message based on the Processing Role. The response is stored in the responseDetails element.<br>• Timeout: This algorithm detects whether a communication has been waiting for an asynchronous response too long. |
| Retry | The lifecycle state responsible for transitioning the outbound communication back to Awaiting Response. |
| Response Error | The business object enters the Response Error state when an error has occurred in sending the request. It contains algorithms for creating To Do entries and for retrying similar to the activity’s Communication Error state. |
| Completed | This is the final state representing a successful outbound communication. Finishing tasks such as setting flags for use by subsequent communications and notifying the parent activity of completion are performed. |
| Discarded | Includes an algorithm that transfers the parent activity to Failed. |

## Inbound Communications

Inbound communication business objects are based on the D1-COMMIN maintenance objects. Inbound Communication objects have a much less defined schema since they are driven by the needs of the head end system’s API and those of the business process. As with outbound


Copyright © 2026, Oracle and/or its affiliates.

communications, the schema does not represent the exact message, but is the post-processed XSL result.

### Key Algorithms

This section outlines some of the important algorithms executed as part of the inbound communication's lifecycle.

Some algorithms are directly shared by multiple business objects and multiple lifecycle states. In other cases, unique algorithms will perform the same role across business objects. The following section refers to the algorithms generically.

| Lifecycle State | Algorithms |
|---|---|
| Validate | Similar to the activity, the Validate state houses algorithms that will execute as the message is created.<br>• Validate Communication Type: This algorithm ensures the communicationType field is populated on the message.<br>• Find Parent Outbound Communication for Async Inbound: This algorithm is responsible for linking the incoming message with the original outbound communication using, for example, a transaction ID. This algorithm is what causes an inbound communication to be linked with a larger process. |
| Validation Error | If the Validate state encounters an error, the business object transitions to this state. The algorithms are similar to those in the activity business object. |
| Evaluate Response | The Evaluate Response state processes the inbound communication in the context of an "asynchronous response." The algorithms should perform any required update tasks such as setting flags on the parent outbound communication. |
| Completed | This successful end state for inbound communications should transition the parent outbound communication to the next state. This typically completes the outbound communication and may trigger the activity to create a subsequent outbound communication. |
| Discarded | The parent outbound communication should transition to a failure state. This will also place the activity into the Communication Error state where it will await corrections or a Retry attempt. |

## Synchronous Commands

Synchronous commands, also known as "request-reply" commands, are single HTTP transactions with an initial request followed by a response. The request will contain whatever information is necessary to complete the command, such as meter identifiers or dates. The reply will contain any results generated by the command. It could be a simple acknowledgement or a complex data structure. In general, it will return quickly as HTTP transactions will fail if the response does not arrive quickly enough.

- Example: Itron Commissioning Process
- Itron Commissioning XSL Transformations

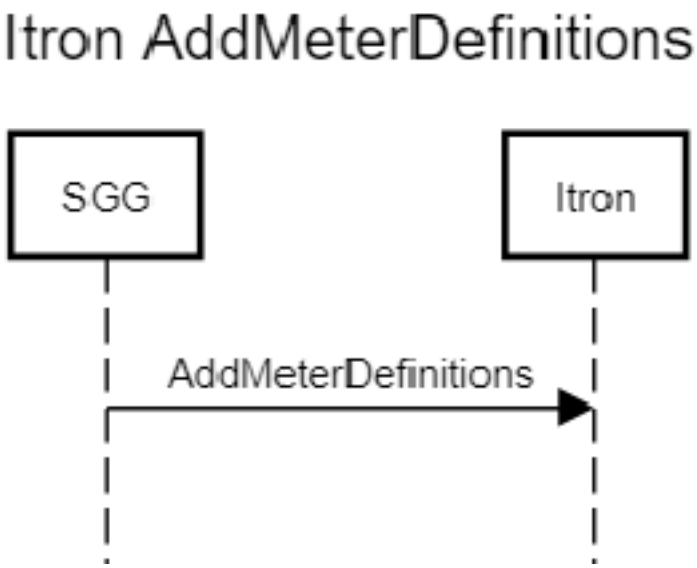
## Example: Itron Commissioning Process

This section provides an example of a synchronous command, the commission device command supported by the Itron OpenWay adapter.


Copyright © 2026, Oracle and/or its affiliates.

The Itron OpenWay 3.9 command for registering a meter (commissioning a device) in the system is AddMeterDefinitions in the `www.itron.com.ami.2008.10.provisioning.wsdl` definition. The Smart Grid Gateway Adapter for Itron OpenWay typically sends a meter serial number and some other configuration data. The response indicates whether or not an error was encountered.

The image below illustrates the process flow used by this command.




Below is a summary of the important features of the Smart Grid Gateway implementation of this command. Many of the concepts can be applied when creating a custom synchronous command.

**Command Activity — Device Commissioning**

Business Object: D1-DeviceCommissioning

**Lifecycle and Algorithms**

The table below outlines some of the important algorithms used by the Device Commissioning command activity.

| Lifecycle State | Algorithms |
|---|---|
| Validate | The important features here are not the specific algorithms, but rather the ability to perform validation logic at a process level.<br>• D1-VALDVCNAC – Validate Device Not Already Commissioned: This algorithm checks whether the device is already commissioned<br>• D1-VHCPCOMMS – Validate Head-End's Capability to Commission Device: This algorithm checks to ensure that the head-end system has the capability to support the command requested. Some commands may not be supported by every head end system.<br>• D1-CACTCOMM – Check for Existing Active Commissioning Command Request: This algorithm prevents multiple, simultaneous commands from being issued by SGG.<br>• D1-CHKFDCOMM – Check for Concurrent Decommissioning Command Request: This algorithm prevents a commissioning request from executing when there is a current decommissioning request. |


Copyright © 2026, Oracle and/or its affiliates.

| Lifecycle State | Algorithms |
|---|---|
| Commission Ready | Several algorithms in this state are designed to create outbound communications to the head end system based on various conditions.<br>• D1-AMIOBCOMM – AMI Device Identifier Outbound Communication Creation: This algorithm creates an outbound message, a communication to the head end system. |
| Communication In Progress | • D1-CHKCHILD – Check Children Communications: This algorithm checks all of the outbound communications related to the activity and, if complete, will transition the activity. |
| Execute Completion Events | • D1-EXCMPEVTS – Execute Completion Events: This algorithm provides a mechanism for running completion events upon successful completion of the commissioning operation. |

### Outbound Communication — Itron - Add Meter Definition (Commission)

Business Object: D8-AddMeterDefinitionsDR

#### Schema

The portions of the schema that contain the payload of the web service request and response communications are the `sendDetail` element (constructed using the D8-AddMeterDefinitionsDRDA data area) and the `responseDetail` element (constructed using the D8-AddMeterDefRespDRDA data area). The XSLs used to create or decipher XML conforming to the Itron structure use these elements as a basis.

#### Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the Add Meter Definition outbound communication.

| Lifecycle State | Algorithms |
|---|---|
| Validate | • D1-VALCOMTP – Validate Communication Type: This algorithm ensures the communication type is populated on the communication. |
| Awaiting Response | • D8-PAMDSDDR – Itron - Populate Add Meter Definitions Send Detail - Direct Route: This algorithm constructs the `sendDetails` element used in the outbound request.<br>• D8-COUTMCMDR – Create Outbound Message for Commissioning / Decommissioning Direct Route: This algorithm determines the correct outbound message to create based upon the input processing role. The reply from the head end system will be stored in the `responseDetails` element and log entries will be created for the creation of the outbound message and the result (whether it is a success or a failure). |
| Response Error | The business object will transition to this state if the reply indicates a failure.<br>• D8-RAMBOE – Itron - Retry Add Meter BO in Error: This algorithm provides a means to retry the communication in the event of an error.<br>• D8-CTDEBOEO – Itron - Create To Do Entry for BO in Error for Outbound: This algorithm creates a To Do entry in order to inform a user to take corrective action.<br>• D1-ALLOWTPA – Set Allow Transition Parent Activity Flag To True For Synchronous Outbound Communication: This algorithm conditionally allows the parent activity to transition to complete. |


Copyright © 2026, Oracle and/or its affiliates.

| Lifecycle State | Algorithms |
|---|---|
| Create Completion Event | • D8-CRCDCOMPE – Itron - Create Commission Device Completion Event: This algorithm creates a completion event to show the device as commissioned. |
| Completed | • D8-UCCFTPA – Itron - Update Commission Completion Flag: This algorithm notifies the parent command activity that the communication has completed. |

#### Message Sender XSLs

Outbound messages created by Add Meter Definition outbound communication use the following XSL files:

- **Request XSL**: D8-CommissionRequest.xsl
- **Response XSL**: D8-CommissionResponse.xsl

Refer to Creating Message Senders for more information about creating message senders.

## Itron Commissioning XSL Transformations

The Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay supports multiple versions of the Itron OpenWay API. To achieve this support, several of the request XSLs check a version number embedded in the source schema. Similarly, the responses XSLs are designed to handle different source XML and map the values to the common SGG format.

The Itron commissioning process uses the following XSL transformations.

#### D8-CommissionRequest.xsl

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:serArr="http://schemas.microsoft.com/2003/10/Serialization/
Arrays"
        xmlns:prov390="http://www.itron.com/ami/2009/08/provisioning"
        xmlns:prov370="http://www.itron.com/ami/2008/10/provisioning">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
        <xsl:apply-templates select="./requestEnvelope/AddMeterDefinitions"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.70']">
        <xsl:apply-templates select="." mode="Itr370"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.90']">
        <xsl:apply-templates select="." mode="Itr390"/>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr370">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/provisioning">
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <xsl:apply-templates select="*|text()" mode="Itr370"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr390">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2009/08/provisioning">
            <xsl:apply-templates select="*|text()" mode="Itr390"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr390">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr370">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

**D8-CommissionResponse.xsl**

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:serArr="http://schemas.microsoft.com/2003/10/Serialization/
Arrays"
        xmlns:prov390="http://www.itron.com/ami/2009/08/provisioning"
        xmlns:prov370="http://www.itron.com/ami/2008/10/provisioning">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
        <xsl:apply-templates select="./requestEnvelope/AddMeterDefinitions"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.70']">
        <xsl:apply-templates select="." mode="Itr370"/>
    </xsl:template>

    <xsl:template match="AddMeterDefinitions[../itronVersion='3.90']">
        <xsl:apply-templates select="." mode="Itr390"/>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr370">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/provisioning">
```


Copyright © 2026, Oracle and/or its affiliates.

```
            <xsl:apply-templates select="*|text()" mode="Itr370"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="*[local-name() != 'string']" mode="Itr390">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2009/08/provisioning">
            <xsl:apply-templates select="*|text()" mode="Itr390"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr390">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="string" mode="Itr370">
        <xsl:element name="string" namespace="http://schemas.microsoft.com/
2003/10/Serialization/Arrays">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

## Asynchronous Commands

An asynchronous command is actually a collection of two or more synchronous commands. This type of command is useful when more than a few seconds will elapse before results can be collected and returned. A typical use case will begin with a Smart Grid Gateway request to start a job on the head end system. The initial request will contain enough data to initiate the process, such as a meter identifier or a date range. The response usually contains an identifier for the process started on the head end system. At some later time, a request will be initiated by the head end system. The request will contain the same process identifier so it can be linked to the original request. Some head end systems use this secondary request to indicate the status of the process. Other head end systems send the results of the process in the secondary request. Typically, the response message for either type of secondary request is a functional acknowledgement.

- Example: Itron On-Demand Read Process
- Itron On-Demand Read XSL Transformations

## Example: Itron On-Demand Read Process

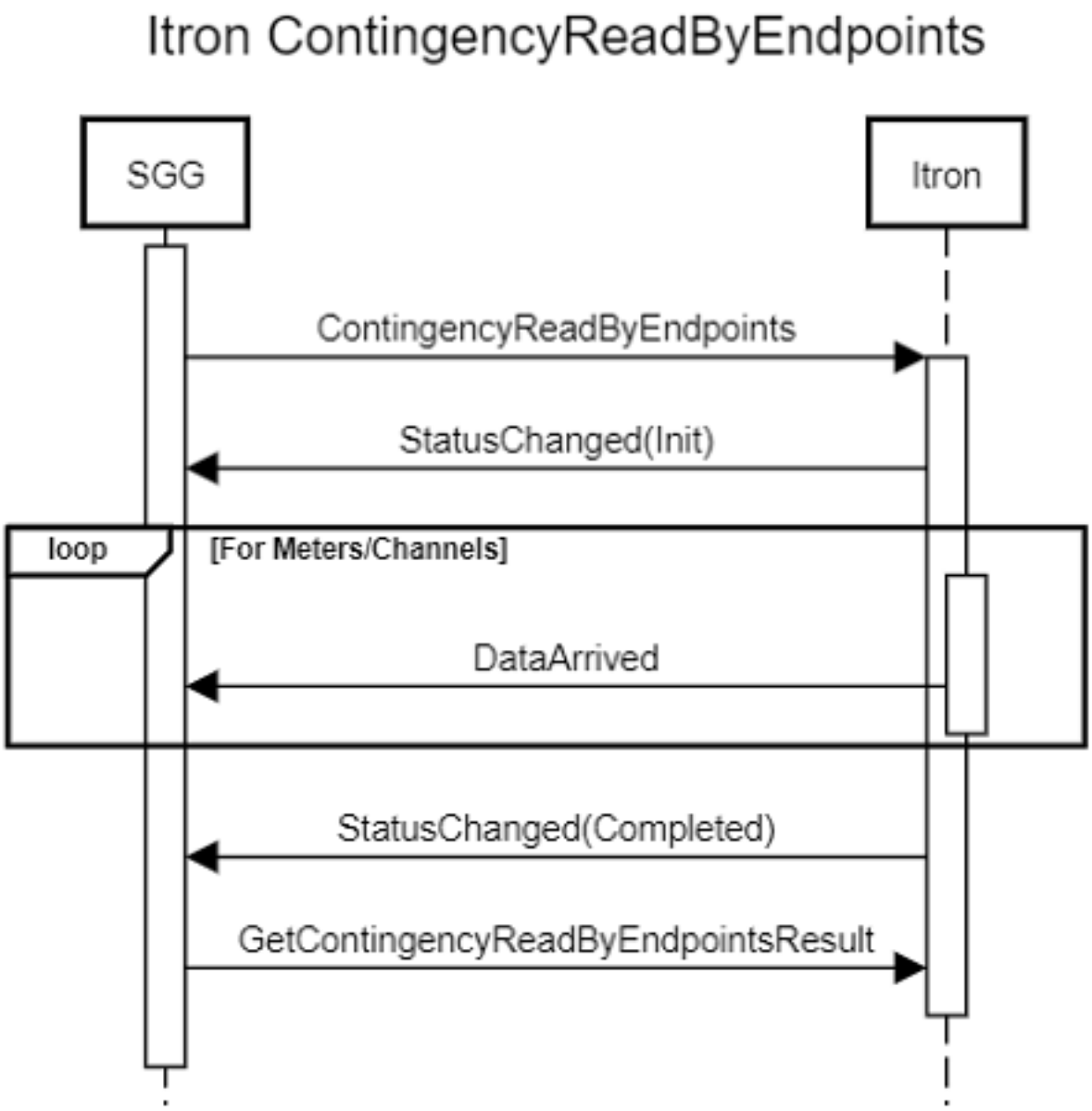
This section provides an example of an asynchronous command, the On-Demand Read command supported by the Itron OpenWay adapter.

The Itron `www.itron.com.ami.2008.10.data.wsdl` definition contains a series of commands for an On-Demand Read process. The Smart Grid Gateway adapter for Itron OpenWay initiates the process by invoking the ContingencyReadByEndpoints method. The reply contains a “request token ID” that identifies the entire flow and will be shared by subsequent communications. On the Itron side, a job for collecting usage and events is


Copyright © 2026, Oracle and/or its affiliates.

configured. To inform the caller (SGG) that the job is starting, Itron sends a StatusChanged request (defined in `www.itron.com.ami.2008.10.common.wsdl`) containing “State = Initialized”. Itron sends another StatusChanged request to notify SGG that the process of gathering the usage is underway. As the usage is collected, Itron submits one or more DataArrived requests to SGG. When Itron has finished making internal requests for usage, it sends a final StatusChanged request to SGG. That completion notification informs SGG that it should submit GetContingencyReadByEndpointsResult, the reply to which contains information about errors encountered during the process.

The image below illustrates the process flow used by this command.




Below is a summary of the important features of the Smart Grid Gateway implementation of this command. Many of the concepts can be applied when creating a custom synchronous command.

**Command Activity — On-Demand Read (Interval)**

**Business Object**: D1-OnDemandReadInterval


Copyright © 2026, Oracle and/or its affiliates.

### Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the On-Demand Read (Interval) command activity.

| Lifecycle State | Algorithms |
|---|---|
| Validate | The important features here are not the specific algorithms, but rather the ability to perform validation logic at a process level.<br>• D1-VALDEVMC – Validate that Device has appropriate Measuring Component: This algorithm ensures the device has at least one measuring component whose Measuring Component Type is registered as interval type<br>• D1-VALIDCMD – Validate Headend's capability to perform On Demand Read (Interval): This algorithm checks whether the head end system has the capability to support an on-demand read request. If the command is supported, an outbound communication business object is returned.<br>• D1-CHKMST – Check for existing Measurements: This algorithm checks whether complete measurements for the specified time period exist in the database already. |
| Connection Ready | • D1-CODRIOB – Create Outbound communication for OnDemandRead Interval: This algorithm creates an outbound communication business object based on the Processing Role and algorithm parameters. The algorithm will look up the outbound communication business object defined for the Processing Role parameter and create an instance of that business object. In the case of this business object, the parameter is D1IN and it should create an instance of the D8-ReadIntervalDR business object.<br>Once the Outbound Communication record is created, the Activity business object moves from the Connection Ready state to the Communication In Progress state. When the outbound communication is complete, it moves the Activity to next default state, which is Connection Ready (the activity moves from Communication In Progress *back* to Connection Ready). The outbound communication will also set the “isOnDemandReadComplete” flag value in the activity business object, in this case, indicating the outbound communication for a particular processing role was created. If the condition is true, the processing moves on to next algorithm.<br>• D1-CODRSRTOB – Create On Demand Read Results Outbound based on Completion Flag: This algorithm is the second algorithm in this state responsible for sending outbound communications. If the completion flag passed as a parameter is true, it will create the second outbound communication. In this example, it should create an instance of D8-ReadOnDemandReadResultDR to send a second request to Itron for the results of the Contingency Read operation. |
| Waiting for Measurement | • D1-RIINTM – Retrieve Interval Initial Measurements: This algorithm retrieves the most recent Initial Measurement Data for each interval measuring component for the period of the measurement requested. |

### Outbound Communication — Itron - Contingency Read (Interval)

**Business Object**: D8-ReadIntervalDR

**Schema**


Copyright © 2026, Oracle and/or its affiliates.

The portions of the schema that contain the payload of the web service request and response communications are the `sendDetail` element (constructed using the D8-ContingencyReadRequestDR data area) and the `responseDetail` element (constructed using the D8-ContingencyReadResponDtlDA data area). The XSLs used to create or decipher XML conforming to the Itron structure use these elements as a basis.

### Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the Contingency Read outbound communication.

| Lifecycle State | Algorithms |
|---|---|
| Validate | • D1-VALCOMTP – Validate Communication Type: This algorithm ensures the communication type is populated on the communication. |
| Awaiting Response | • D8-PSDMRSCDR – Itron - Populate Send Detail for ContingencyReadByEndpoints Direct Route: This algorithm populates the `sendDetail` element with the needed items for the web service input.<br>• D8-COUTMINDR – Itron - Create Outbound Message Contingency Read (Interval): This algorithm determines the correct outbound message to create based upon the input processing role. The reply from the head end system will be stored in the `responseDetails` element and log entries will be created for the creation of the outbound message and the result (whether it is a success or a failure). |
| Response Error | • D8-RBORINTER – Itron - Retry Read Interval BO in Error: This algorithm provides a means to retry the communication in the event of an error. |
| Completed | • D1-UCFLAGTPA – Update On-Demand Read Completion Flag And Transition Parent Activity: If requirements are met, this algorithm sets the completion flag on the parent activity to “true” and transitions it to the next state. |

## Inbound Communication — Itron - StatusChanged

**Business Object**: D8-StatusChanged

This business object handles the inbound StatusUpdate requests. It contains the request token ID created by the initial request. When it receives schema values of “State = Completed” and Result of either Canceled or Success, it triggers the next command in the Activity. For On-DemandRead, the next command is GetContingencyReadByEndpointsResult.

### Schema

Unlike other communications containing `sendDetail` or `responseDetail` elements, this business object contains an input element with child elements that map to the those within the StatusChanged XSD definition.

### Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the Status Changed Read inbound communication.


Copyright © 2026, Oracle and/or its affiliates.

| Lifecycle State | Algorithms |
|---|---|
| Validate | • D8-IDNPOCAA – Find Parent Outbound Communication For Async Inbound: This algorithm finds the initial outbound communication associated with the current inbound communication using the request token id. |
| Evaluate Response | • D8-EVALSTCHG – Itron - Evaluate Response for StatusChanged: This algorithm performs a task based on the values of the arriving StatusChanged request. If the Status is Completed and the Result is Success or Failure, the parent outbound communication is completed. This alerts the activity to initiate the next outbound communication. |

### Outbound Communication — Itron - On Demand Read Result

**Business Object**: D8-ReadOnDemandReadResultDR

This is the second of two outbound requests within the Itron Contingency Read API. The outbound request submits the request token ID. The response contains error information collected by the Itron server during the execution of the Contingency Read process.

#### Schema

The portions of the schema that contain the payload of the web service request and response communications are the `sendDetail` element (constructed using the D8-ReadODRResultRequestDR data area) and the `responseDetail` element (constructed using the D8-ReadODRResultResponseDR data area). The XSLs used to create or decipher XML conforming to the Itron structure use these elements as a basis.

#### Lifecycle and Algorithms

The table below outlines some of the important algorithms used by the On-Demand Read Result outbound communication.

| Lifecycle State | Algorithms |
|---|---|
| Awaiting Response | • D8-PSDODRRDR – Itron - Populate OnDemand Read Result Send Details: This algorithm populate the `sendDetail` element for the outbound message. Primarily, this consists of the request token ID.<br>• D8-CODRROBDR – Create Outbound Message for On Demand Read Result: This algorithm determines the correct outbound message to create based upon the input Processing Role. The reply from the head end system will be stored in the `responseDetails` element and log entries will be created for the creation of the outbound message and the result (whether it is a success or a failure). |
| Evaluate Response | • D8-EVODRRSLT – Create Outbound Message for On Demand Read Result: This algorithm evaluates the response from Itron and determines whether the command was successful or was a failure.<br>• D8-VDVCMCDR – Validate if Device has appropriate Measuring Component: This algorithm performs a check to ensure that the reading has a proper measuring component. |


Copyright © 2026, Oracle and/or its affiliates.

#### Service Script: Itron - DataArrived Request

This Groovy—based service script handles the incoming DataArrived requests. It saves the payload to file storage locations (including Cloud Object Storage buckets in the case of cloud implementations) based on configuration.

#### Schema

The schema maps closely to the Itron DataArrived message. It includes the D8-DataArrived data area.

#### Steps

#### Step 10 Edit Data – Invoke Main Groovy Method

This step shows how to trigger a Groovy script from other OUAF objects. It is similar to a main method in other programming languages. In this case, it is called by an Inbound Web Service method.

#### Step 20 – Groovy Imports

This step imports other classes needed by the script.

#### Step 30 – Groovy Members

1. Look up the Master Configuration object
2. Use the Master Configuration to find the File Storage details
    a. Find the File Adapter type
    b. Compute a filename using a configured generation script
    c. Compute a fill URL using the bucket, filename, and other details
3. Use the FileStorageAdapter business service to save the file to the targeted location.

### Message Sender XSLs

Outbound messages created by on-demand read outbound communications use the following XSL files:

#### ContingencyReadByEndpoints

- **Request XSL**: D8-ContinegencyReadByEndpointsRequest.xsl
- **Response XSL**: D8-ContingencyReadByEndpointsResponse.xsl

#### GetContingencyReadByEndpointsResult

- **Request XSL**: D8-GetContingencyReadByEndpointsResultRequest.xsl
- **Response XSL**: D8-GetContingencyReadByEndpointsResultResponse.xsl

Refer to Creating Message Senders for more information about creating message senders.

### Inbound Web Services

The Itron on-demand read process uses the following inbound web services.

#### D8-RequestStatusChangedService

- **Operation Name**: StatusChanged
- **Schema Name**: Itron — Status Changed BO


Copyright © 2026, Oracle and/or its affiliates.

- **Request XSL**: D8-RequestStatusChangedServiceRequest.xsl
- **Response XSL**: D8-RequestStatusChangedServiceResponse.xsl

**D8-DataSubscriberService**

- **Operation Name**: DataArrived
- **Schema Name**: Itron — DataArrived Request Script
- **Request XSL**: D8-DataSubscriberServiceRequest.xsl
- **Response XSL**: D8-DataSubscriberServiceResponse.xsl

## Itron On-Demand Read XSL Transformations

The Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay supports multiple versions of the Itron OpenWay API. To achieve this support, several of the request XSLs check a version number embedded in the source schema. Similarly, the responses XSLs are designed to handle different source XML and map the values to the common SGG format.

The Itron on-demand read process uses the following XSL transformations.

**D8-ContRdByEndptsRequest.xsl**

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:ns2="http://www.itron.com/ami/2008/10/common"
        xmlns:ns3="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
       <ContingencyReadByEndpoints xmlns="http://www.itron.com/ami/2008/10/
data">
           <xsl:apply-templates select="./requestEnvelope/
ContingencyReadByEndpoints/*"/>
       </ContingencyReadByEndpoints>
    </xsl:template>


    <xsl:template match="ElectronicSerialNumbers">
        <xsl:element name="ns2:ElectronicSerialNumbers">
            <xsl:for-each select="./string">
                <xsl:element name="ns3:string">
                    <xsl:value-of select="."/>
                </xsl:element>
            </xsl:for-each>
        </xsl:element>
    </xsl:template>


    <xsl:template match="StatusChangedService">
        <xsl:element name="ns2:StatusChangedService">
            <xsl:value-of select="."/>
        </xsl:element>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    </xsl:template>


    <xsl:template match="Parameters">
        <ns1:Parameters>
            <ns1:ReadingStartTime><xsl:value-of select="./
ReadingStartTime"/></ns1:ReadingStartTime>
            <ns1:ReadingEndTime><xsl:value-of select="./ReadingEndTime"/></
ns1:ReadingEndTime>
            <xsl:if test="string-length(./PerformDemandReset) > 0">
                <ns1:PerformDemandReset><xsl:value-of select="./
PerformDemandReset"/></ns1:PerformDemandReset>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveHomeNetworkData) > 0">
                <ns1:RetrieveHomeNetworkData><xsl:value-of select="./
RetrieveHomeNetworkData"/></ns1:RetrieveHomeNetworkData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveInstantaneousData) > 0">
                <ns1:RetrieveInstantaneousData><xsl:value-of select="./
RetrieveInstantaneousData"/></ns1:RetrieveInstantaneousData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveLastDemandReset) > 0">
                <ns1:RetrieveLastDemandReset><xsl:value-of select="./
RetrieveLastDemandReset"/></ns1:RetrieveLastDemandReset>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveLoadProfileData) > 0">
                <ns1:RetrieveLoadProfileData><xsl:value-of select="./
RetrieveLoadProfileData"/></ns1:RetrieveLoadProfileData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveLogEvents) > 0">
                <ns1:RetrieveLogEvents><xsl:value-of select="./
RetrieveLogEvents"/></ns1:RetrieveLogEvents>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveNetworkStatistics) > 0">
                <ns1:RetrieveNetworkStatistics><xsl:value-of select="./
RetrieveNetworkStatistics"/></ns1:RetrieveNetworkStatistics>
            </xsl:if>
            <xsl:if test="string-length(./RetrievePriorSelfRead) > 0">
                <ns1:RetrievePriorSelfRead><xsl:value-of select="./
RetrievePriorSelfRead"/></ns1:RetrievePriorSelfRead>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveRecentRegisters) > 0">
                <ns1:RetrieveRecentRegisters><xsl:value-of select="./
RetrieveRecentRegisters"/></ns1:RetrieveRecentRegisters>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveVoltageMonitorData) > 0">
                <ns1:RetrieveVoltageMonitorData><xsl:value-of select="./
RetrieveVoltageMonitorData"/></ns1:RetrieveVoltageMonitorData>
            </xsl:if>
            <xsl:if test="string-length(./ReportSelfReadOption) > 0">
                <ns1:ReportSelfReadOption><xsl:value-of select="./
ReportSelfReadOption"/></ns1:ReportSelfReadOption>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveExtendedCurrentData) > 0">
                <ns1:RetrieveExtendedCurrentData><xsl:value-of select="./
RetrieveExtendedCurrentData"/></ns1:RetrieveExtendedCurrentData>
```


Copyright © 2026, Oracle and/or its affiliates.

```
            </xsl:if>
            <xsl:if test="string-length(./RetrievePriorExtendedSelfReadData)
> 0">
                <ns1:RetrievePriorExtendedSelfReadData><xsl:value-of
select="./RetrievePriorExtendedSelfReadData"/></
ns1:RetrievePriorExtendedSelfReadData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveInstrumentationProfileData)
> 0">
                <ns1:RetrieveInstrumentationProfileData><xsl:value-of
select="./RetrieveInstrumentationProfileData"/></
ns1:RetrieveInstrumentationProfileData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveExtendedLoadProfileData) >
0">
                <ns1:RetrieveExtendedLoadProfileData><xsl:value-of select="./
RetrieveExtendedLoadProfileData"/></ns1:RetrieveExtendedLoadProfileData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveToolboxData) > 0">
                <ns1:RetrieveToolboxData><xsl:value-of select="./
RetrieveToolboxData"/></ns1:RetrieveToolboxData>
            </xsl:if>
            <xsl:if test="string-length(./RetrieveTemperatureMonitoringData)
> 0">
                <ns1:RetrieveTemperatureMonitoringData><xsl:value-of
select="./RetrieveTemperatureMonitoringData"/></
ns1:RetrieveTemperatureMonitoringData>
            </xsl:if>
        </ns1:Parameters>
    </xsl:template>


    <xsl:template match="*">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/data">
            <xsl:apply-templates select="*|text()"/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

**D8-ContRdByEndptsResponse.xsl**

```
<xsl:stylesheet
        version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        exclude-result-prefixes="env ns1 com">

<xsl:output method="xml" encoding="utf-8" indent="yes"/>

<xsl:template match="env:Envelope">
   <xsl:apply-templates select="./env:Body/*"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xsl:template>


<xsl:template match="ns1:ContingencyReadByEndpointsResponse">
    <responseDetail>
       <responseEnvelope>
           <ContingencyReadByEndpointsResponse>
               <xsl:apply-templates select="./*"/>
           </ContingencyReadByEndpointsResponse>
       </responseEnvelope>
    </responseDetail>
</xsl:template>


<xsl:template match="env:Fault[./detail/*[contains('|OperationFault|
NoValidTargetsFoundFault|InvalidUriFormatFault|InvalidDateTimeRangeFault|
ArgumentFault|TooManyConcurrentRequestsFault|SecurityFault|',
concat('|',local-name(),'|'))]]">
    <fault>
        <xsl:apply-templates select="./detail/*"/>
    </fault>
</xsl:template>


<xsl:template match="env:Fault">
    <fault>
        <xsl:apply-templates select="./*"/>
    </fault>
</xsl:template>


<xsl:template match="detail">
    <detail><xsl:apply-templates select="./*" mode="serialize"/></detail>
</xsl:template>


<xsl:template match="*">
    <xsl:variable name="name">
       <xsl:value-of select="local-name()"/>
   </xsl:variable>
   <xsl:element name="{$name}" namespace="">
       <xsl:apply-templates select="*|text()"/>
   </xsl:element>
</xsl:template>


<xsl:template match="*" mode="serialize">
    <xsl:text><</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:for-each select="@*">
        <xsl:value-of select="concat(' ', local-name())"/>
        <xsl:text>="</xsl:text>
        <xsl:value-of select="."/>
        <xsl:text>"</xsl:text>
    </xsl:for-each>
    <xsl:text>></xsl:text>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    <xsl:apply-templates select="*" mode="serialize"/>
    <xsl:value-of select="."/>
    <xsl:text></</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:text>></xsl:text>
</xsl:template>

</xsl:stylesheet>
```

**D8-GetContRdByEndptsRsReq.xsl**

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:ns2="http://www.itron.com/ami/2008/10/common">

    <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="yes"
indent="yes"/>

    <xsl:template match="/*">
      <GetContingencyReadByEndpointsResult xmlns="http://www.itron.com/ami/
2008/10/data">
         <xsl:apply-templates select="./requestEnvelope/
GetContingencyReadByEndpointsResult/*" />
      </GetContingencyReadByEndpointsResult>
    </xsl:template>

    <xsl:template match="Id">
        <xsl:element name="ns2:Id">
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="*">
        <xsl:element name="{local-name()}" namespace="http://
www.itron.com/ami/2008/10/data">
            <xsl:apply-templates select="*|text()"/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

**D8-GetContRdByEndptsRsResp.xsl**

```
<xsl:stylesheet
        version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:ns1="http://www.itron.com/ami/2008/10/data"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        exclude-result-prefixes="env ns1 com">

<xsl:output method="xml" encoding="utf-8" indent="yes"/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
<xsl:template match="env:Envelope">
    <xsl:apply-templates select="./env:Body/*"/>
</xsl:template>

<xsl:template match="ns1:GetContingencyReadByEndpointsResultResponse">
    <responseDetail>
       <responseEnvelope>
           <GetContingencyReadByEndpointsResultResponse>
               <xsl:apply-templates select="./*"/>
           </GetContingencyReadByEndpointsResultResponse>
       </responseEnvelope>
    </responseDetail>
</xsl:template>

<xsl:template match="*">
    <xsl:variable name="name">
        <xsl:value-of select="local-name()"/>
    </xsl:variable>
    <xsl:element name="{$name}" namespace="">
        <xsl:apply-templates select="*|text()"/>
    </xsl:element>
</xsl:template>

<xsl:template match="env:Fault[./detail/*[contains('|OperationFault|
RequestNotFoundFault|RequestNotFinishedFault|ArgumentFault|
TooManyConcurrentRequestsFault|SecurityFault|', concat('|',local-
name(),'|'))]]">
    <fault>
        <xsl:apply-templates select="./detail/*"/>
    </fault>
</xsl:template>


<xsl:template match="env:Fault">
    <fault>
        <xsl:apply-templates select="./*"/>
    </fault>
</xsl:template>


<xsl:template match="detail">
    <detail><xsl:apply-templates select="./*" mode="serialize"/></detail>
</xsl:template>

<xsl:template match="*" mode="serialize">
    <xsl:text><</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:for-each select="@*">
        <xsl:value-of select="concat(' ', local-name())"/>
        <xsl:text>="</xsl:text>
        <xsl:value-of select="."/>
        <xsl:text>"</xsl:text>
    </xsl:for-each>
    <xsl:text>></xsl:text>
    <xsl:apply-templates select="*" mode="serialize"/>
    <xsl:value-of select="."/>
```


Copyright © 2026, Oracle and/or its affiliates.

```
    <xsl:text></</xsl:text>
    <xsl:value-of select="local-name()"/>
    <xsl:text>></xsl:text>
</xsl:template>

</xsl:stylesheet>
```

**D8-DataSubSvcRequest.xsl**

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:soap12="http://www.w3.org/2003/05/soap-envelope"
        xmlns:data="http://www.itron.com/ami/2008/10/data"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        xmlns:ev="http://www.itron.com/ami/2008/10/events"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        exclude-result-prefixes="data com ev soap11 soap12">

    <xsl:template match="soap11:Envelope">
        <xsl:apply-templates select="//data:DataArrived"/>
    </xsl:template>


    <xsl:template match="soap11:Body">
        <xsl:apply-templates select="//data:DataArrived"/>
    </xsl:template>


    <xsl:template match="data:DataArrived">
        <DataArrived>
            <input>
                <RequestToken>
                    <Id><xsl:value-of select="./data:input/data:RequestToken/
com:Id"/></Id>
                </RequestToken>
                <ReadDataCollection>
                    <xsl:for-each select="./data:input/
data:ReadDataCollection/data:ReadData">
                        <ReadData>
                            <Identifier><xsl:value-of select="./
data:Identifier"/></Identifier>
                            <DeviceClass><xsl:value-of select="./
data:DeviceClass"/></DeviceClass>
                            <ScalingType><xsl:value-of select="./
data:ScalingType"/></ScalingType>
                            <xsl:call-template name="EventLog"/>
                            <xsl:call-template name="LoadProfileChannels"/>
                            <xsl:call-template name="NetworkStatistics"/>
                            <xsl:call-template name="RegisterValues"/>
                            <xsl:call-template name="BlockInfo"/>
                            <xsl:if test="./data:DeviceSerialNumber">
                                <DeviceSerialNumber><xsl:value-of select="./
data:DeviceSerialNumber"/></DeviceSerialNumber>
                            </xsl:if>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                        </ReadData>
                    </xsl:for-each>
                </ReadDataCollection>
                <xsl:if test="count(./data:input/data:RequestMetadata/*) > 0">
                    <RequestMetadata>
                        <xsl:copy-of select="./data:input/
data:RequestMetadata/*"/>
                    </RequestMetadata>
                </xsl:if>
            </input>
        </DataArrived>
    </xsl:template>


    <xsl:template name="EventLog">
        <xsl:if test="count(./data:EventLog/ev:MeterEvents/*) > 0 or count(./
data:EventLog/ev:UnreportedEventCount) > 0">
            <EventLog>
                <xsl:if test="count(./data:EventLog/ev:MeterEvents/*) > 0">
                    <MeterEvents>
                        <xsl:for-each select="./data:EventLog/ev:MeterEvents/
ev:MeterEvent">
                            <MeterEvent>
                                <Timestamp><xsl:value-of select="./
ev:Timestamp"/></Timestamp>
                                <Category><xsl:value-of select="./
ev:Category"/></Category>
                                <LogType><xsl:value-of select="./
ev:LogType"/></LogType>
                                <Name><xsl:value-of select="./ev:Name"/></
Name>
                                <xsl:if test="count(./ev:Arguments/*) > 0">
                                    <Arguments>
                                        <xsl:for-each select="./ev:Arguments/
ev:Argument">
                                            <Argument>
                                                <Name><xsl:value-of select="./
ev:Name"/></Name>
                                                <Value><xsl:value-of
select="./ev:Value"/></Value>
                                            </Argument>
                                        </xsl:for-each>
                                    </Arguments>
                                </xsl:if>
                                <xsl:if test="./ev:ID">
                                    <ID><xsl:value-of select="./ev:ID"/></ID>
                                </xsl:if>
                                <xsl:if test="./ev:Source">
                                    <Source><xsl:value-of select="./
ev:Source"/></Source>
                                </xsl:if>
                                <xsl:if test="./ev:SequenceNumber">
                                    <SequenceNumber><xsl:value-of select="./
ev:SequenceNumber"/></SequenceNumber>
                                </xsl:if>
                            </MeterEvent>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    </xsl:for-each>
                </MeterEvents>
            </xsl:if>
            <xsl:if test="./data:EventLog/ev:UnreportedEventCount">
                <UnreportedEventCount><xsl:value-of select="./
data:EventLog/ev:UnreportedEventCount"/></UnreportedEventCount>
            </xsl:if>
        </EventLog>
    </xsl:if>
</xsl:template>


<xsl:template name="LoadProfileChannels">
    <xsl:if test="count(./data:LoadProfileChannels/
data:LoadProfileChannel/data:IntervalLength) > 0">
        <LoadProfileChannels>
            <xsl:for-each select="./data:LoadProfileChannels/
data:LoadProfileChannel">
                <LoadProfileChannel>
                    <IntervalLength><xsl:value-of select="./
data:IntervalLength"/></IntervalLength>
                    <PulseMultiplier><xsl:value-of select="./
data:PulseMultiplier"/></PulseMultiplier>
                    <Quantity><xsl:value-of select="./data:Quantity"/></
Quantity>
                    <TimeDataEnd><xsl:value-of select="./
data:TimeDataEnd"/></TimeDataEnd>
                    <IntervalValues>
                        <xsl:for-each select="./data:IntervalValues/
data:IntervalValue">
                            <IntervalValue>
                                <ChannelValue><xsl:value-of select="./
data:ChannelValue"/></ChannelValue>
                                <xsl:for-each select="./
data:ProfileStatuses/data:ProfileStatus">
                                    <ProfileStatuses>
                                        <ProfileStatus><xsl:value-of
select="."/></ProfileStatus>
                                    </ProfileStatuses>
                                </xsl:for-each>
                            </IntervalValue>
                        </xsl:for-each>
                    </IntervalValues>
                </LoadProfileChannel>
            </xsl:for-each>
        </LoadProfileChannels>
    </xsl:if>
</xsl:template>


<xsl:template name="NetworkStatistics">
    <xsl:if test="count(./data:NetworkStatistics/data:NetworkStatistic/
data:Name) > 0">
        <NetworkStatistics>
            <xsl:for-each select="./data:NetworkStatistics/
data:NetworkStatistic">
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <NetworkStatistic>
                        <Name><xsl:value-of select="./data:Name"/></Name>
                        <NetworkKind><xsl:value-of select="./
data:NetworkKind"/></NetworkKind>
                        <Value><xsl:value-of select="./data:Value"/></Value>
                    </NetworkStatistic>
                </xsl:for-each>
            </NetworkStatistics>
        </xsl:if>
    </xsl:template>


    <xsl:template name="RegisterValues">
        <xsl:if test="count(./data:RegisterValues/data:RegisterValue/
data:Quantity) > 0">
            <RegisterValues>
                <xsl:for-each select="./data:RegisterValues/
data:RegisterValue">
                    <RegisterValue>
                        <Quantity><xsl:value-of select="./data:Quantity"/></
Quantity>
                        <Source><xsl:value-of select="./data:Source"/></
Source>
                        <Timestamp><xsl:value-of select="substring(./
data:Timestamp, 1, 19)"/></Timestamp>
                        <Value><xsl:value-of select="./data:Value"/></Value>
                        <xsl:if test="count(./data:Status/data:string) > 0">
                            <Status>
                                <xsl:for-each select="./data:Status/
data:string">
                                    <string><xsl:value-of select="."/></
string>
                                </xsl:for-each>
                            </Status>
                        </xsl:if>
                    </RegisterValue>
                </xsl:for-each>
            </RegisterValues>
        </xsl:if>
    </xsl:template>


    <xsl:template name="BlockInfo">
        <xsl:if test="count(./data:BlockInfo/data:Block) > 0">
            <BlockInfo>
                <Block><xsl:value-of select="./data:BlockInfo/data:Block"/></
Block>
                <TotalBlocks><xsl:value-of select="./data:BlockInfo/
data:TotalBlocks"/></TotalBlocks>
            </BlockInfo>
        </xsl:if>
    </xsl:template>


    <!-- When the SOAP Message has already been transformed by the SOAP Proxy
-->
```


Copyright © 2026, Oracle and/or its affiliates.

```
    <xsl:template match="DataArrived[namespace-uri(.) != 'http://
www.itron.com/ami/2008/10/data']">
        <xsl:copy-of select="."/>
    </xsl:template>

</xsl:stylesheet>
```

**D8-DataSubSvcResponse.xsl**

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:tns="http://www.itron.com/ami/2008/10/data">

    <xsl:template match="/">
        <soap11:Envelope>
            <soap11:Header/>
            <soap11:Body>
                <tns:DataArrivedResponse/>
            </soap11:Body>
        </soap11:Envelope>
    </xsl:template>

</xsl:stylesheet>
```

**D8-ReqStsChgSvcRequest.xsl**

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:soap12="http://www.w3.org/2003/05/soap-envelope"
        xmlns:com="http://www.itron.com/ami/2008/10/common"
        exclude-result-prefixes="soap11 soap12 com">

    <xsl:template match="soap11:Envelope">
        <xsl:apply-templates select="//com:StatusChanged"/>
    </xsl:template>


    <xsl:template match="soap11:Body">
        <xsl:apply-templates select="//com:StatusChanged"/>
    </xsl:template>


    <xsl:template match="com:StatusChanged">
        <StatusChanged>
            <bo/>
            <input>
                <RequestStatus>
                    <RequestToken>
                        <Id><xsl:value-of select="./com:input/
com:RequestStatus/com:RequestToken/com:Id"/></Id>
                    </RequestToken>
                    <State><xsl:value-of select="./com:input/
com:RequestStatus/com:State"/></State>
```


Copyright © 2026, Oracle and/or its affiliates.

```
                    <EndpointsScheduled><xsl:value-of select="./com:input/
com:RequestStatus/com:EndpointsScheduled"/></EndpointsScheduled>
                    <Result><xsl:value-of select="./com:input/
com:RequestStatus/com:Result"/></Result>
                    <SuccessfulEndpoints><xsl:value-of select="./com:input/
com:RequestStatus/com:SuccessfulEndpoints"/></SuccessfulEndpoints>
                    <FailedEndpoints><xsl:value-of select="./com:input/
com:RequestStatus/com:FailedEndpoints"/></FailedEndpoints>
                    <xsl:if test="string-length(./com:input/com:RequestStatus/
com:CreatedWhen) > 0">
                        <CreatedWhen><xsl:value-of select="substring(./
com:input/com:RequestStatus/com:CreatedWhen, 1, 23)"/></CreatedWhen>
                    </xsl:if>
                    <xsl:if test="string-length(./com:input/com:RequestStatus/
com:FinishedWhen) > 0">
                        <FinishedWhen><xsl:value-of select="substring(./
com:input/com:RequestStatus/com:FinishedWhen, 1, 23)"/></FinishedWhen>
                    </xsl:if>
                    <xsl:if test="count(./com:input/com:RequestStatus/
com:Metadata/*) > 0">
                        <Metadata>
                            <xsl:copy-of select="./com:input/
com:RequestStatus/com:Metadata/*"/>
                        </Metadata>
                    </xsl:if>
                </RequestStatus>
            </input>
        </StatusChanged>
    </xsl:template>


    <!-- When the SOAP Message has already been transformed by the SOAP Proxy
-->
    <xsl:template match="StatusChanged[namespace-uri(.) != 'http://
www.itron.com/ami/2008/10/common']">
        <xsl:copy-of select="."/>
    </xsl:template>

</xsl:stylesheet>
```

**D8-RequestStatusChangedServiceResponse.xsl**

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:tns="http://www.itron.com/ami/2008/10/common">

    <xsl:template match="/">
        <soap11:Envelope>
            <soap11:Header/>
            <soap11:Body>
                <tns:StatusChangedResponse/>
            </soap11:Body>
        </soap11:Envelope>
    </xsl:template>
```


Copyright © 2026, Oracle and/or its affiliates.

```
</xsl:stylesheet>
```

# Cloud Service Test Harness

The Smart Grid Gateway Cloud Service Test Harness is used to simulate a head end system for testing smart meter commands in a cloud implementation.

The test harness is "stateless", meaning it does not store meters or their current status. The harness is designed to return a successful result to all command requests by default. Devices must be created in a specific manner to return an error.

### How the Test Harness Works

The test harness includes SOAP web service implementations for each command supported by the various Smart Grid Gateway adapters. The Test Harness's container is managed by Oracle cloud services and internal routing definitions allow web service requests to locate the proper container without requiring users to update ports or other configurations. There is no user-accessible configuration required for the test harness itself.

The test harness uses a specific system-owned user (D1SGG) to send asynchronous communications back to Smart Grid Gateway.

### Creating Devices for use with the Test Harness

The test harness works with any sort of device capable to sending smart meter commands. By default, all commands are successful.

Devices must be created in a specific manner in order for errors to be returned from the test harness. Specifically, commands for devices whose identifiers (badge number, serial number, internal meter number, pallet number, external ID, MDM external ID, and asset ID) end in "ERR" will return an error. This means that when setting up devices for testing via the test harness, you must create separate devices for successful and unsuccessful (error) results.

The simplest approach when creating "error" devices is to use the same identifier value for all identifiers, but at a minimum, "ERR" MUST be the final part of the identifier value for the **AMI Device ID Type** and **AMI Pre-Commissioning Device ID Type** configured on the head end system.

### Defining Head End Systems, Processing Methods, Outbound Message Types, and Message Senders for use with the Test Harness

This section provides guidelines for creating data and records used with smart meter commands when using the test harness.

#### Head End System

When configuring a head end system for use with the test harness, note the identifier types specified in the **AMI Device ID Type** and **AMI Pre-Commissioning Device ID Type** fields. When creating devices intended to return an error, the identifier values for the identifier type(s) specified for these fields MUST end in "ERR" to return an error from the test harness.

#### Outbound Message Types

When creating outbound message types for use with the test harness, use the appropriate outbound business object based on the specific command. See **Smart Meter Command Flows** for details.

#### Message Senders


Copyright © 2026, Oracle and/or its affiliates.

When creating message senders for use with the test harness, use the following settings:

**Main** tab:

- **Invocation Type**: Real-Time
- **Message Class**: SOAPSNDR
- **Active**: <checked>
- **Msg Encoding**: UTF-8 message encoding

**Context** tab:

- **HTTP Header**: soapAction: <soapaction>

  where <soapaction> is the URI (Soap Action) for the command as shown under **Adapter-Specific Test Harness Commands** below.

  Note the format used here is:

  SoapAction: {URI}

  That's using a **colon, followed by a space** between the SoapAction and the value. You **cannot** use an equals (=) sign. For example:

  ```
  SoapAction: http://www.multispeak.org/Version_4.1_Release/
  MeterAddNotification
  ```

- **HTTP Login User**: A user which has been assigned to the SGGHarnessUser role in Oracle Identity Cloud Service (IDCS). See **Setting Up a User with Access to the Smart Grid Gateway Test Harness** in the *Oracle Utilities Cloud Services Administration Guide* for more information.
- **HTTP Login Password**: The password for the user above.
- **HTTP Method**: POST
- **HTTP URL 1**: <endpoint_URL>

  where <endpoint_URL> is the HTTP URL or the command as shown under **Adapter-Specific Test Harness Commands** below, formatted as follows:

  ```
  @SGG_TEST@<adapter_name>/<webservice>
  ```

  where:

  - @SGG_TEST@ is a token that combines the protocol, host, port, tenant, domain, and application
  - <adapter_name> is one “lg”, “itron”, “ssn”, or “sensus”, as appropriate
  - <webservice> is a name specific to each adapter (e.g. OD_Server, MR_CB, DeviceManager, and so on. See Adapter-Specific Test Harness Commands)

> **Note**
>
> When creating message senders for use with the test harness, do **NOT** specify a **Sender Security Type** on the **Context** tab.

### External Systems

When creating outbound message types on external systems and for use with the test harness, be sure to specify the appropriate Message XSL and Response XSL files based on the outbound message business object. See Creating an External Systemfor lists of these files.


Copyright © 2026, Oracle and/or its affiliates.

### Processing Methods

Command processing methods for a head end system used with the test harness should reference an Outbound Message Type, either the **Default Outbound Message Type** or an **Outbound Message Type** for a specific **Device Type**.

- Adapter-Specific Test Harness Commands

## Adapter-Specific Test Harness Commands

This section provides specific details for invoking commands for specific adapters.

Each adapter comprises multiple web services that enable commands. The commands available for each web service for each adapter are provided below.

### HTTP Header Format

The format used for the HTTP Header is as follows:

SoapAction: {URI}

That's using a **colon, followed by a space** between the SoapAction and the value. You **cannot** use an equals (=) sign. For example:

```
SoapAction: http://www.multispeak.org/Version_4.1_Release/
MeterAddNotification
```

### Itron OpenWay Commands

### Control Service

The table below lists details for commands available via the Itron OpenWay Control Service web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Connect | ReconnectMeter | SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/ReconnectMeter | @SGG_TEST@/itron/www.itron.com.ami.2008.10.control |
| Connect (Result) | GetReconnectMeterResult | SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/GetReconnectMeterResult | @SGG_TEST@/itron/www.itron.com.ami.2008.10.control |
| Disconnect | DisconnectMeter | SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/DisconnectMeter | @SGG_TEST@/itron/www.itron.com.ami.2008.10.control |
| Disconnect (Result) | GetDisconnectMeterResult | SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/GetDisconnectMeterResult | @SGG_TEST@/itron/www.itron.com.ami.2008.10.control |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Load Check | DetectLoadSideVoltageByMeter | SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/DetectLoadSideVoltageByMeter | @SGG_TEST@/itron/www.itron.com.ami.2008.10.control |
| Load Check (Result) | GetDetectLoadSideVoltageByMeterResult | SoapAction: http://www.itron.com/ami/2008/10/control/ControlService/GetDetectLoadSideVoltageByMeterResult | @SGG_TEST@/itron/www.itron.com.ami.2008.10.control |

### Control Diagnostic

The table below lists details for commands available via the Itron OpenWay Control Diagnostic web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Device Status Check | PingByEndpoints | SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/PingByEndpoints | @SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic |
| Device Status Check (Result) | GetPingByEndpointsResult | SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/GetPingByEndpointsResult | @SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic |
| Read Disconnect State | ReadDisconnectStateByMeters | SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/ReadDisconnectStateByMeters | @SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic |
| Read Disconnect State (Result) | GetReadDisconnectStateByMetersResult | SoapAction: http://www.itron.com/ami/2012/03/control/diagnostic/DiagnosticService/GetReadDisconnectStateByMetersResult | @SGG_TEST@/itron//www.itron.com.ami.2012.03.control.diagnostic |

### Data Service

The table below lists details for commands available via the Itron OpenWay Data Service web service.


Copyright © 2026, Oracle and/or its affiliates.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| On-Demand Read | ContingencyReadByEndpoints | SoapAction: http://www.itron.com/ami/2008/10/data/DataService/ContingencyReadByEndpoints | @SGG_TEST@/itron//www.itron.com.ami.2008.10.data |
| On-Demand Read (Result) | GetContingencyReadByEndpointsResult | SoapAction: http://www.itron.com/ami/2008/10/data/DataService/GetContingencyReadByEndpointsResult | @SGG_TEST@/itron//www.itron.com.ami.2008.10.data |
| Scheduled Read | InterrogateByGroup | SoapAction: http://www.itron.com/ami/2008/10/data/DataService/InterrogateByGroup | @SGG_TEST@/itron//www.itron.com.ami.2008.10.data |
| Scheduled Read (Result) | GetInterrogateByGroupResult | SoapAction: http://www.itron.com/ami/2008/10/data/DataService/GetInterrogateByGroupResult | @SGG_TEST@/itron//www.itron.com.ami.2008.10.data |

### Provisioning Service

The table below lists details for commands available via the Itron OpenWay Provisioning Service web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Commission | AddMeterDefinitions | SoapAction: http://www.itron.com/ami/2009/08/provisioning/ProvisioningService/AddMeterDefinitions | @SGG_TEST@/itron/www.itron.com.ami.2009.08.provisioning |
| Decommission | DeregisterMeters | SoapAction: http://www.itron.com/ami/2009/08/provisioning/ProvisioningService/DeregisterMeters | @SGG_TEST@/itron/www.itron.com.ami.2009.08.provisioning |

### Landis+Gyr Commands

#### MR_CB

The table below lists details for commands available via the Landis+Gyr MR_CB web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Commission | MeterAddNotification | SoapAction: http://www.multispeak.org/Version_3.0/MeterRemoveNotification | @SGG_TEST@/lg/MR_CB |


Copyright © 2026, Oracle and/or its affiliates.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Decommission | MeterRemoveNotification | SoapAction: http://www.multispeak.org/Version_3.0/MeterAddNotification | @SGG_TEST@/lg/MR_CB |
| Meter Retire | MeterRetireNotification | SoapAction: http://www.multispeak.org/Version_3.0/MeterRetireNotification | @SGG_TEST@/lg/MR_CB |
| On-Demand Read | InitiateMeterReadByMeterNumber | SoapAction: http://www.multispeak.org/Version_3.0/InitiateMeterReadByMeterNumber | @SGG_TEST@/lg/MR_CB |
| Scheduled Demand Reset | ScheduleDemandReset | SoapAction: http://www.multispeak.org/Version_3.0/ScheduleDemandReset | @SGG_TEST@/lg/MR_CB |

### CD_MR

The table below lists details for commands available via the Landis+Gyr CD_MR web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Meter Exchange | MeterExchangeNotification | SoapAction: http://www.multispeak.org/Version_3.0/MeterExchangeNotification | @SGG_TEST@/lg/CB_MR |

### CD_CB

The table below lists details for commands available via the Landis+Gyr CD_CB web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Connect/Disconnect | InitiateConnectDisconnect | SoapAction: http://www.multispeak.org/Version_3.0/InitiateConnectDisconnect | @SGG_TEST@/lg/CD_CB |

### Metering

The table below lists details for commands available via the Landis+Gyr Metering web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Add Meter to Inventory | AddMeterToInventory | SoapAction: turtletech.com/TS2/AddMeterToInventory | @SGG_TEST@/lg/Metering |

### CIM

The table below lists details for commands available via the Landis+Gyr CIM web service.


Copyright © 2026, Oracle and/or its affiliates.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| On Demand Read (Interval & Scalar)<br>Device Status Check | AMIRequest | SoapAction: urn:LandisGyr.GSIS.CimService.CIMService.AMIRequest | @SGG_TEST@/lg/CIMService |

**Note**: The Landis+Gyr ScheduleDemandReset API does not offer a way to pass a callback URL in. In actual use, L+G allows the callback to be configured in their system. To handle the scenario in the Test Harness, add an additional HTTP header named **SGG-Callback-Endpoint** to the callback URL in the Message Sender. The value of the header should be the address of the D3-CB_MR Inbound Web Service, for example: SGG-Callback-Endpoint: “https://144.25.14.142:8086/iddev2/prod/ccs/soap/api/iws/D3-CB_MR”. The URL value is located near the bottom of the WSDL on the Inbound Web Service record.

### Sensus Commands

#### MR_Server

The table below lists details for commands available via the Sensus MR_Server web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Commission | MeterAddNotification | SoapAction: http://www.multispeak.org/Version_4.1_Release/MeterAddNotification | @SGG_TEST@/sensus/MR_Server |
| Decommission | MeterRemoveNotification | SoapAction: http://www.multispeak.org/Version_4.1_Release/MeterRemoveNotification | @SGG_TEST@/sensus/MR_Server |
| On-Demand Read | InitiateMeterReadingsByMeterNumber | SoapAction: http://www.multispeak.org/Version_4.1_Release/InitiateMeterReadingsByMeterID | @SGG_TEST@/sensus/MR_Server |

#### CD_Server

The table below lists details for commands available via the Sensus CD_Server web service.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Connect/Disconnect | InitiateConnectDisconnect | SoapAction: http://www.multispeak.org/Version_4.1_Release/InitiateConnectDisconnect | @SGG_TEST@/sensus/CD_Server |

#### OD_Server

The table below lists details for commands available via the Sensus OD_Server web service.


Copyright © 2026, Oracle and/or its affiliates.

| Command | Web Method | HTTP Header | HTTP URL |
|---|---|---|---|
| Device Status Check | InitiateOutageDetectionEventRequest | SoapAction: http://www.multispeak.org/Version_4.1_Release/InitiateOutageDetectionEventRequest | @SGG_TEST@/sensus/OD_Server |

### Silver Spring Networks Commands

Note: The Silver Spring Networks adapter works differently from the other adapters. This adapter does not use SoapAction identifiers on the operations, so the HTTP Header can be left off of the Message Sender configuration.

#### DataAggregation

The table below lists details for commands available via the Silver Spring Networks DataAggregation web service.

| Command | Web Method | HTTP URL |
|---|---|---|
| | GetMeterFieldStatus | @SGG_TEST@/ssn/DataAggregation |

#### DeviceManager

The table below lists details for commands available via the Silver Spring Networks DeviceManager web service.

| Command | Web Method | HTTP URL |
|---|---|---|
| All | FindDevice | @SGG_TEST@/ssn/DeviceManager |
| Commission | ReplaceLocation | @SGG_TEST@/ssn/DeviceManager |
| Decommission | ReplaceDeviceAtLocation | @SGG_TEST@/ssn/DeviceManager |

#### JobManager

The table below lists details for commands available via the Silver Spring Networks JobManager web service.

| Command | Web Method | HTTP URL |
|---|---|---|
| Connect/Disconnect | addRemoteProvisioningJob | @SGG_TEST@/ssn/JobManager |
| On-Demand Read | addMeterReadJob | @SGG_TEST@/ssn/JobManager |
| Device Status Check | addPingJob | @SGG_TEST@/ssn/JobManager |
| On-Demand Read, Device Status Check | getJobStatus | @SGG_TEST@/ssn/JobManager |
| Connect/Disconnect | getJobStatusForDevice | @SGG_TEST@/ssn/JobManager |

#### DeviceResults

The table below lists details for commands available via the Silver Spring Networks DeviceResults web service.


Copyright © 2026, Oracle and/or its affiliates.

| Command | Web Method | HTTP URL |
|---|---|---|
| Connect/Disconnect | getRemoteProvisioningResults ByJobID | @SGG_TEST@/ssn/ DeviceResults |
| On-Demand Read | getMeterReadResultsByJobID | @SGG_TEST@/ssn/ DeviceResults |
| Device Status Check | getPingresultsByJobID | @SGG_TEST@/ssn/ DeviceResults |

# Web Service Authentication

This section describes configuration settings for web service authentication for each Smart Grid Gateway Adapter. Each adapter section below includes guidelines for defining credentials to outbound web service requests and instructions for delivering credentials when calling back into the application via inbound web services.

For the purposes of this document, “Inbound” refers to requests originating from outside Smart Grid Gateway and are targeted at Smart Grid Gateway web services. “Outbound” requests originate within Smart Grid Gateway and are targeted at web services hosted on other servers.

- Itron OpenWay Web Service Authentication
- Landis+Gyr Web Service Authentication
- Sensus Web Service Authentication
- Silver Spring Networks Web Service Authentication
- Adapter Development Kit Web Service Authentication

## Itron OpenWay Web Service Authentication

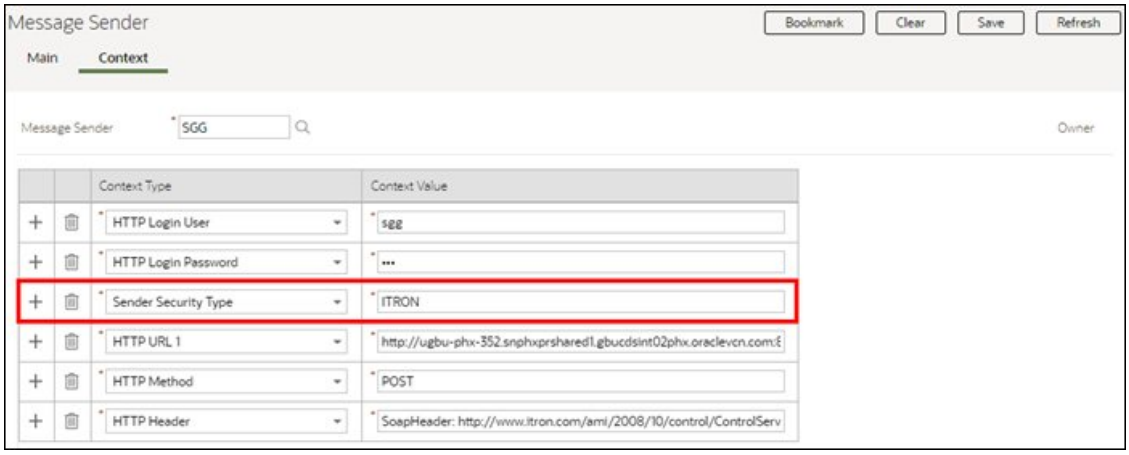
### Inbound Requests

Incoming requests should utilize standard basic HTTPS authentication.

### Outbound Requests

The Itron system can be configured to expect a WSSE security element in the SOAP header for authentication. That element will contain certain features such as a nonce, in addition to the credentials. This can be configured in Smart Grid Gateway by setting the Message Sender’s **Sender Security Type** value to "ITRON" (as shown in the following screen capture).


Copyright © 2026, Oracle and/or its affiliates.




## Landis+Gyr Web Service Authentication

The L+G Adapter communicates using the MultiSpeak 3.0 standard as well as the Common Information Model (CIM) standard. The MultiSpeak standard includes a SOAP Header element, MultiSpeakMsgHeader with "UserId" and "Pwd" security attributes. The Message Sender configuration is used to insert credentials into the header element. Incoming requests will be scanned for header credentials when an instruction to do so is found in the URL.

Similarly, the CIM standard defines credential locations within the SOAP Header and Body, depending on the version.

**MultiSpeak 3.0**

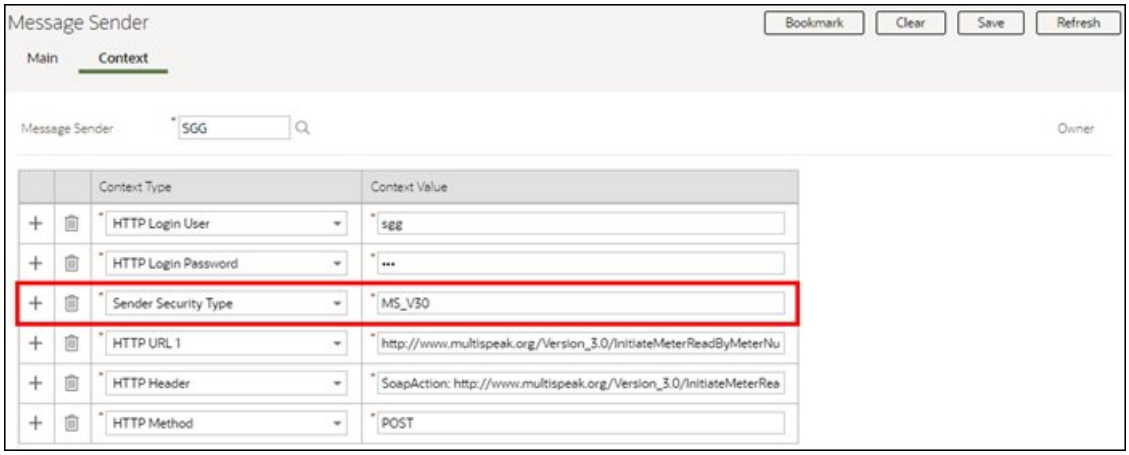
**Inbound Requests**

Include “/multispeak30” prior to the inbound web service name in the URL to signal the application that it should locate credentials in the MultiSpeakMsgHeader SOAP Header element.

Example: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/multispeak30**/D3-CB_MR

**Outbound Requests**

On the **Context** tab of the **Message Sender** portal, add a **Sender Security Type** row with a value of “MS_V30” (as shown in the following screen capture). The values of the **HTTP Login User** and **HTTP Login Password** will be added as attributes of MultiSpeakMsgHeader in the outgoing SOAP header.





Copyright © 2026, Oracle and/or its affiliates.

#### CIM 1.2

**Inbound Requests**

Include “/cim12” prior to the inbound web service name in the URL to signal the application that it should include SOAP Body credentials ({Specific Message Type}/Header/User/UserId and {Specific Message Type}//Header/Property/Name[text() = 'Password']).

Example: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/cim12**/D3-CIMCallback

**Outbound Requests**

CIM 1.2 outbound requests should utilize either standard basic HTTPS authentication or send credentials in the SOAP Header element as in CIM 2.0.

#### CIM 2.0

**Inbound Requests**

Include “/cim20” prior to the inbound web service name in the URL to signal the application that it should include credentials in the SOAP Header in the form of elements:

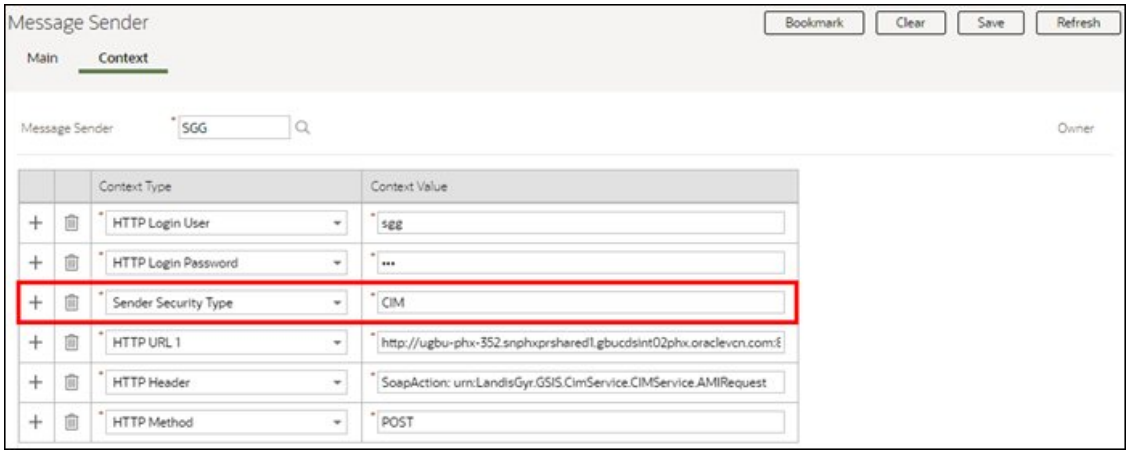
- {http://www.landisgyr.com/iec61968/2010/03}UserName
- {http://www.landisgyr.com/iec61968/2010/03}Password

Example: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/cim20**/D3-CIMCallback

**Outbound Requests**

On the **Context** tab of the **Message Sender** portal, add a **Sender Security Type** row with a value of “CIM” (as shown in the following screen capture). The values of the **HTTP Login User** and **HTTP Login Password** will be added to the outgoing SOAP Header at as elements:

- {http://www.landisgyr.com/iec61968/2010/03}UserName
- {http://www.landisgyr.com/iec61968/2010/03}Password




## Sensus Web Service Authentication

#### MultiSpeak 4.1 Release

**Inbound Requests**

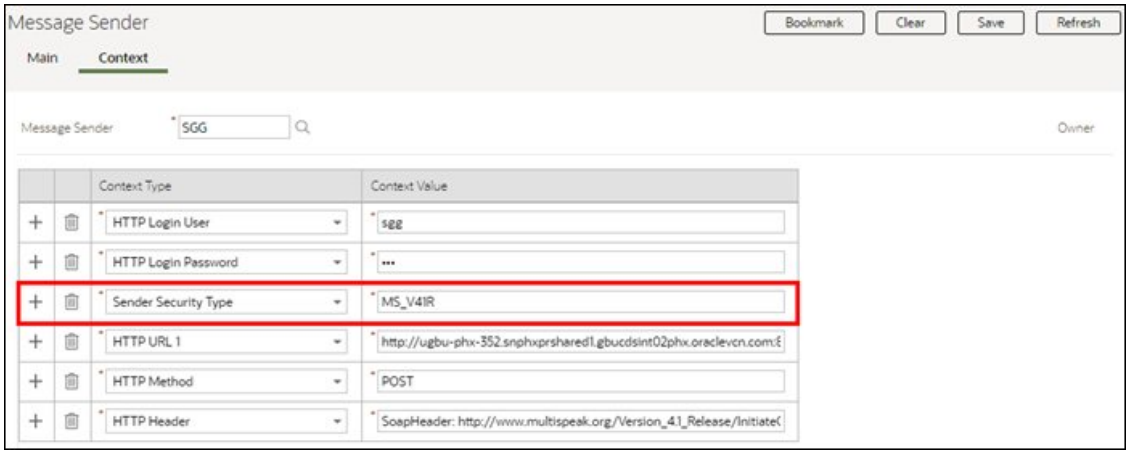

Copyright © 2026, Oracle and/or its affiliates.

Include “/multispeak41” prior to the inbound web service name in the URL to signal the application that it should locate credentials in the MultiSpeakMsgHeader SOAP Header element.

Ex: https://144.25.13.161:8086/tnt22a/test03/ccs/soap/api/iws**/multispeak41**/D6-CB_Server

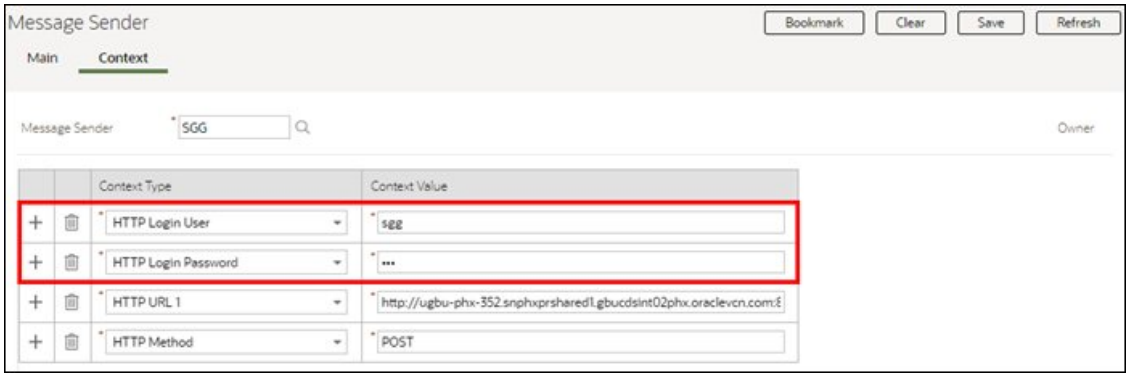
**Outbound Requests**

On the **Context** tab of the **Message Sender** portal, add a **Sender Security Type** row with a value of “MS_V41R” (as shown in the following screen capture). The values of the **HTTP Login User** and **HTTP Login Password** will be added as attributes of MultiSpeakMsgHeader in the outgoing SOAP header.




## Silver Spring Networks Web Service Authentication

The Silver Spring Neworks AMM API does not support Inbound Requests (from the SilverSprings server to Smart Grid Gateway). Outbound requests should utilize standard basic HTTPS authentication options via the **HTTP Login User** and **HTTP Login Password** parameters on the **Context** tab of the **Message Sender** portal.




## Adapter Development Kit Web Service Authentication

**Inbound Requests**

The Adapter Development Kit utilizes standard basic HTTPS authentication for inbound requests.


Copyright © 2026, Oracle and/or its affiliates.

### Outbound Requests

Outbound requests should utilize standard basic HTTPS authentication options the **HTTP Login User** and **HTTP Login Password** parameters on the **Context** tab of the **Message Sender** portal.

# Master Configurations

Master configurations define configuration settings for individual adapters. The sections below provide details for the master configuration provided for each Smart Grid Gateway adapter.

### Itron OpenWay Master Configuration

The “Itron Configuration” master configuration defines details used by the Itron OpenWay commands and payload processing, including the following:

- **Itron Version**: Used to specify the version of the Itron OpenWay head end system being used.
- **Command Status Changed URL Configuration**: Used to define URL configuration for asynchronous callback requests.
  - **Status Changed Service URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Status Changed Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This base URL is appended by command-specific URL components defined as algorithm parameters on "Populate Send Details." algorithms on outbound communication business objects.
  - **Commands**: Command-specific base URLs for asynchronous callback requests
- **Arrived Data Save Locations**: Used to define how payloads and exceptions are routed and processed by the Itron DataSubscriberService, Itron EventSubscriberService, and Itron ExceptionSubscriberService inbound web services.
  - **DataArrived Output Configuration**: Defines how payloads are routed and processed by the Itron DataSubscriberService inbound web service.


Copyright © 2026, Oracle and/or its affiliates.

        - **Save to Staging**: A flag that indicates if payloads routed and processed by the Itron DataSubscriberService inbound web service should be saved to a staging area in Oracle Object Storage. This should be set to "Yes" to enable the use of a staging area in Object Storage.
        - **File Storage**: The staging location where payloads are routed for later processing. This should be a File Location Extendable Lookup value.
        - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where payloads are routed.
        - **Filename Generation Script**: The name of a service script used to compose the filename when saving incoming payloads to the staging area. By default, the D8-DEFFNGEN script is used.
    - **EventsArrived Output Configuration**: Defines how event payloads are routed and processed by the Itron EventSubscriberService inbound web service.
        - **Save to Staging**: A flag that indicates if event payloads routed and processed by the Itron EventSubscriberService inbound web service should be saved to a staging area in Oracle Object Storage. This should be set to "Yes" to enable the use of a staging area in Object Storage.
        - **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.
        - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
        - **Filename Generation Script**: The name of a service script used to compose the filename when saving incoming event payloads to the staging area. By default, the D8-DEFFNGEN script is used.
    - **ExceptionsArrived Output Configuration**: Defines how exceptions are routed and processed by the Itron ExceptionSubscriberService inbound web service
        - **Save to Staging**: A flag that indicates if exceptions routed and processed by the Itron ExceptionSubscriberService inbound web service should be saved to a staging area in Oracle Object Storage. This should be set to "Yes" to enable the use of a staging area in Object Storage.
        - **File Storage**: The staging location where exceptions are routed for later processing. This should be a File Location Extendable Lookup value.
        - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where exceptions are routed.
        - **Filename Generation Script**: The name of a service script used to compose the filename when saving exceptions to the staging area. By default, the D8-DEFFNGEN script is used.

#### Itron Field Collection System Master Configuration

The “FCS Configuration” master configuration defines details used by the Itron FCS payload processing, including the following:

- **File Header Information**: XML tags used in the file header for export files, including XML Declaration, Import, and Workset tags.
- **FCS Mapping Information**: Details regarding mapping of data from FCS to Smart Grid Gateway


Copyright © 2026, Oracle and/or its affiliates.

- **General Configuration**: Characteristic types used to define values for things such as Special Codes, Transport Modes, Tamper Counts, and so on, as well as the Residential Service Point Type and Rate Code used with FCS customers.
- **Walk-by Read Method**: Read methods used when collecting FCS meter reads, such as "Manual" or "Mobile Meter Read".
- **UOM Mapping to Standard UOM Mapping**: Mappings between FCS Text Prompt ("TextPrompt") and combinations of standard units of measure (UOM) and time of use periods (TOU).
- **Read Method Mapping Information**: Mappings between FCS read methods and standard read methods defined in the system.

#### Landis+Gyr Master Configuration

The “L+G – Master Config” master configuration defines details used by the Landis+Gyr commands and payload processing, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This base URL is appended by command-specific URL components defined as algorithm parameters on "Populate Send Details." algorithms on outbound communication business objects.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Arrived Data Save Locations**
  - **CIM 1.0 Events Configuration**: Used to define settings for the CIM 1.0 AMIEvent interface
    - **Save to Staging Area**: Flag that indicate whether to save incoming data to a staging area, such as Oracle Object Storage, or process it directly
    - **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.
    - **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
    - **Filename Generation Script**: A service script used to compose the filename when saving the incoming payload to the staging area. By default, the D8-DEFFNGEN script is used.
  - **CIM 2.0 Events Configuration**: Used to define settings for the CIM 2.0 AMIEvent interface
    - **Save to Staging Area**: Flag that indicate whether to save incoming data to a staging area, such as Oracle Object Storage, or process it directly
    - **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.


Copyright © 2026, Oracle and/or its affiliates.

        * **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
        * **Filename Generation Script**: A service script used to compose the filename when saving the incoming payload to the staging area. By default, the D3-DEFFNGEN script is used.
        * **XSL Transformer Template**: The XSL used when writing data to a file system. By default, the L+G CIM 2.0 - EndDeviceEvents Output (XSL D3-CIMEndDvcEvtsOutput.xsl) xsl is used.
        * **File Extension**: The extension used for the name of the outputted file. It can be specified with or without a leading dot (ex: ".txt" or "xml").

#### Sensus Master Configuration

The “Sensus Configuration” master configuration defines details used by the Sensus commands and outage events, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This base URL is appended by command-specific URL components defined as algorithm parameters on "Populate Send Details." algorithms on outbound communication business objects.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Outage Events Head End System**: Used to define the head end system expected to which outage events are sent via the ODEventNotification interface

#### Silver Spring Networks Version Master Configuration

The “SSN – UIQ Version Configuration” master configuration specifies the version of the Silver Spring Networks head end system with which the system is communicating:

- **SSN Version**: Used to define the version of the Silver Spring Networks head end system. Options include:
  - SSN v4.10 (WSDL 2.1)
  - SSN v4.14 (WSDL 2.5.1)
  - SSN v4.4 (WSDL v1.4)
  - SSN v4.7 (WSDL v1.6)
  - SSN v5.0-5.1 (WSDL v2.7)
  - SSN v5.1 (WSDL v2.8)
  - SSN v5.2 (WSDL v2.9)
  - SSN v6.1 (WSDL v2.11)


Copyright © 2026, Oracle and/or its affiliates.

- **SSN - Get Job Status Configuration**: Used to specify the **Retry Limit** and **Wait Duration (in Seconds)** for the Get Job Status requests sent by outbound communications.

#### Silver Spring Networks MultiSpeak Version Master Configuration

The “SSN – MultiSpeak Master Configuration” master configuration defines details used by the Silver Spring Networks MultiSpeak commands and payload processing, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This URL applies when an override script is not provided and when there is not a business object-specific override configured in the "Command Response URL Configuration" section. If no value is provided, the F1-EnviURL service script is invoked to determine the current URL from the environment variables on the server. Refer to the embedded help for additional details.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Outage Events Head End System**: Used to define the head end system expected to which outage events are sent via the ODEventNotification interface

#### Adapter Development Kit Master Configuration

The “ADK Configuration” master configuration defines details used by Adapter Development Kit commands and outage events, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This URL applies when an override script is not provided and when there is not a business object-specific override configured in the "Command Response URL Configuration" section. If no value is provided, the F1-EnviURL service script is invoked to determine the current URL from the environment variables on the server. Refer to the embedded help for additional details.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Arrived Data Save Locations - Save Locations for Events**: Defines how event payloads are routed and processed.


Copyright © 2026, Oracle and/or its affiliates.

- **File Storage**: The staging location where event payloads are routed for later processing. This should be a File Location Extendable Lookup value.
- **Output Bucket/Folder Name**: The specific object storage bucket or folder within the file storage location where event payloads are routed.
- **Filename Generation Script**: The name of a service script used to compose the filename when saving incoming event payloads to the staging area. By default, the D1DEFFNGEN script is used.
- **XSL Transformer Template**: A Managed Content record (of type XSLTC) used to transform event data
- **File Extension**: The extension used for the name of the outputted file. It can be specified with or without a leading dot (ex: ".txt" or "xml").

#### Adapter Development Kit MultiSpeak Master Configuration

The “ADK MultiSpeak Master Configuration” master configuration defines details used by Adapter Development Kit MultiSpeak commands and outage events, including the following:

- **Command Response URL Configuration**: Used to define URL configuration for asynchronous callback requests
  - **Response URL Script**: Used to define a service script which composes the URL for asynchronous callbacks. The script should include an /output/applicationUrl field containing the results. When determining the URL, the system initially checks if a script has been configured on this field, and if so, that script is used. If this field is blank (which is the default), the Determine Application URL (D1-DetStChSU) script is used to compose the URL.
  - **Default Response Service URL**: The base portion of a URL sent to the head end system which provides a location to which it should send an asynchronous callback request. This URL applies when an override script is not provided and when there is not a business object-specific override configured in the "Command Response URL Configuration" section. If no value is provided, the F1-EnviURL service script is invoked to determine the current URL from the environment variables on the server. Refer to the embedded help for additional details.
  - **Command Response URL Configuration**: Command-specific base URLs for asynchronous callback requests
- **Outage Events Head End System**: Used to define the head end system expected to which outage events are sent via the ODEventNotification interface

## Native Implementation Configuration Checklists

This section provides an overview of the steps involved in configuring Native implementations of Oracle Utilities Smart Grid Gateway adapters, including Payload Processing and Smart Meter Command Processing.

#### Payload Processing Checklist

1. Create Keys and Key Rings (Cloud Only). See Creating Key Rings and Pairs for more information.
2. Configure Object Storage Locations (Cloud Only). See Creating Object Storage Locations for more information.
3. Configure File Storage Extendable Lookup Values. See Creating File Storage Extendable Lookup Values for more information.


Copyright © 2026, Oracle and/or its affiliates.

4. Configure a Head End System to represent the Head End System. See Creating Head End Systems for more information.
5. Configure Processing Methods for the Head End System appropriate for payload processing, including.
    - Initial Measurement Creation
    - Device Event Mapping
    - UOM Translation

    > **Note**
    >
    > These processing methods rely on extendable lookups. See Configuring Adapter Extendable Lookups for more information.
6. Configure SGG Payload Processing Extendable Lookup Values. See Creating SGG Payload Processing Extendable Lookup Values for more information.
7. Configure the adapter's Master Configuration (if appropriate). See Master Configurations for more information.
8. Configure Other Extendable Lookup Values. See Configuring Adapter Extendable Lookups for more information.
9. Configure Batch Controls. See Creating Payload Processing Batch Controls for more information.

**Smart Meter Command Processing Checklist**

1. Configure Activity Types for each command you wish to support in your implementation. Smart meter commands are activities in the system. For each command you plan to support, confirm that you have a corresponding Activity Type configured.
2. Configure Communication Types appropriate for each command you wish to support in your implementation. Each communication sent to the head end system as part of a command request requires a communication type. See the following sections for more information about the specific communications used with each command:
    - Itron OpenWay Command Flows
    - Landis+Gyr Command Flows
    - Sensus Command Flows
    - Silver Spring Networks Command Flows

    > **Note**
    >
    > Note: Most Adapters use outbound communications only.
3. Configure Message Senders for each command you wish to support in your implementation. See Sending Outbound Communications and Creating Message Senders for more information.
4. Configure Outbound Message Types each command you wish to support in your implementation. See Sending Outbound Communications and Creating Outbound Message Types for more information.
5. Configure an External System to represent the Head End System.


Copyright © 2026, Oracle and/or its affiliates.

a. Configure Message Senders/Outbound Message Types for each command you wish to support in your implementation See Sending Outbound Communications and Creating an External System for more information.

6. Configure a Head End System to represent the Head End System. Head End Systems reference the corresponding External System. See Creating Head End Systems for more information.
7. Configure Processing Methods for the Head End System appropriate for each command you wish to support in your implementation. See Configuring Processing Methods for more information.
8. Configure Inbound Web Services as appropriate for the Head End System See Receiving Inbound Communications and Inbound Web Services for more information.
9. Configure the adapter's Master Configuration (if appropriate) See Master Configurations for more information.
10. Configure Extendable Lookups See Configuring Adapter Extendable Lookups for more information.

**Configuration Details Reference**

Refer to Knowledge Base article KB523753 on **My Oracle Support** for a spreadsheet containing configuration details for the following Adapters:

- Itron OpenWay
- Silver Spring Networks
- Landis+Gyr
- MV90 (Payload Processing Only)

This spreadsheet contains lists of configuration objects used by each adapter, including:

- Activity Types
- Communication Types
- Message Senders
- Head End System
- Outbound Message Type
- External System
- Master Configurations (as applicable)
- Extendable Lookups
- Batch Controls

# Defining Service Order Management Options


- Understanding Service Order Management
- Service Order Activities
- Service Order Field Activities
- Field Task Types
- Service Order Management External Applications




Copyright © 2026, Oracle and/or its affiliates.

# Understanding Service Order Management

At a high level, service order management handles requests for service as follows:

**Receive / Create Service Order Request**

A service order request is created and/or received. This can be as a result of a customer requesting a change to their service such as enabling or disabling service when moving into or out of a residence, but can also be the result of other business processes, such as a request to cut service due to non-payment.

Regardless of the origin of the request, a service order request is created in a customer information system (CIS) such as Oracle Utilities Customer Care and Billing), which in turn is sent to Service Order Management.

**Create Service Order Activity**

When Service Order Management receives the service order request, it creates a service order activity. This activity will manage and orchestrate any/all other activities needed to fulfill the original service request.

**Evaluate Service Point and/or Meter**

The service order activity then evaluates the current state of the service point, meter, or item, and determines the appropriate action(s) to take to fulfill the service request.

**Create Activities as Needed / Appropriate**

Based on the evaluation of the service point/meter/item, the service order activity creates one or more activities as needed.

- Service order field activities involve sending workers into the field to perform service. This can include meter installation, meter replacement, and other activities.
- Command activities are smart meter commands used to remotely change the state of the meter. This can include connect, disconnect, checking the device status (ping), or requesting an on-demand reading.

Following each of activities, the orchestration activity re-evaluates the state of the service point/ meter/item to determine the next appropriate action(s) required to fulfill the original service request.

For example, when enabling service at a service point with a smart scalar meter, a typical scenario might involve the following:

1. Service Order Field Activity - Install Meter
2. Command Activity: Commission Device
3. Command Activity: Remote Connect
4. Command Activity: On Demand Read - Scalar

**Update Requesting System**

When the service order activity determines that everything necessary to satisfy the service order request is ready, the service order activity will inform the requester and complete the original request.


Copyright © 2026, Oracle and/or its affiliates.

**Send Notification to Subscribing Systems**

The service order activity can also be configured to send notifications to other subscribing systems regarding the status of the service point/meter/item.

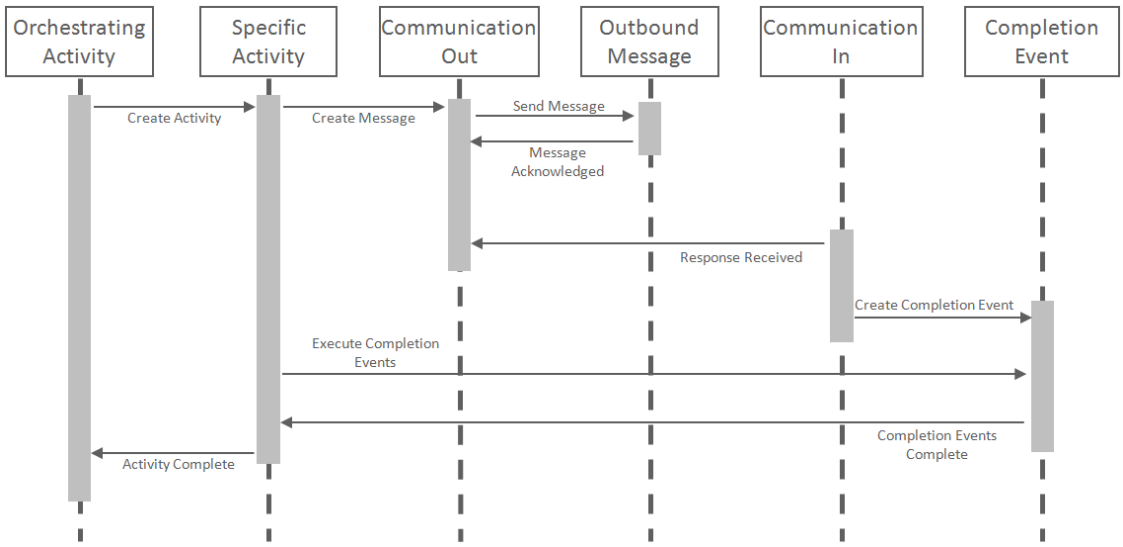
# Service Order Activities

- Understanding Service Order Activities
- Understanding Service Order Activity Types
- Configuring Service Order Activity Types

## Understanding Service Order Activities

This section describes service order activities and how they manage the service order process.

Service order activities coordinate a large number of child transactions that represent the communication to and from an external application. The below diagram depicts a service order activity:




Each object in the sequence diagram has a distinct set of duties within the context of the communication:

- **Orchestrating Activity**: controls the overall intent of the communication. For example it may "Enable Service" which would include initiating one-to-many specific activities to install the meter and begin the flow of the metered commodity to the service point.
- **Specific Activity**: can be initiated from an orchestrating activity or directly. These activities represent a single task to be carried out such as installing a meter or a remote disconnect smart meter command.
- **Communication Out**: orchestrates the communication to the external application and provides robust handling for any errors that might occur during that communication.
- **Outbound Message**: represents the message payload sent to the external system and the synchronous response.


Copyright © 2026, Oracle and/or its affiliates.

- **Communication In**: orchestrates the handling of an asynchronous or unsolicited response from an external system.
- **Completion Event**: carries out the results of the communication. For example, in the case of a remote connect it would create the appropriate on off history entry for the device's installation event.

The base package provides the following types of service order activities:

- **Enable Service**: Used to enable service at a service point.
- **Disable Service**: Used to disable service at a service point.
- **Cut for Non-Payment**: Used to cut off service at a service point due to non-payment of past due amounts.
- **Reconnect Service for Payment**: Used to restore service at a service point after receipt of past due payment.
- **Exchange Meter**: Used to orchestrate the exchange of meters at service point, such as in the event that a customer upgrades their meter.
- **Back to Back Service**: Used to orchestrate a change of service when the customer at a service point changes (such as when owners/tenants change).


- Service Order Activity Processing
- Service Order Activity Algorithm Types


## Service Order Activity Processing

To understand how service order activities manage the service order process, it's important understand the lifecycle of orchestration activity business objects.

### Service Order Orchestration Activity Lifecycle

All service order orchestration activity business objects share a common parent business object that defines their lifecycles. This is the Service Point Activity Orchestration business object (D1-SPActivityOrchestration). The table below outlines the lifecycle for this business object.

| State | Description |
|---|---|
| Pending | The initial state for orchestration activities.<br>An Enter algorithm sends an acknowledgement to the requesting system.<br>The activity is transitioned to the next state via a monitor process. |
| Validate | Enter algorithms perform the following:<br>Validate Activity Type<br>Validate Service Point<br>Check for a non-final duplicate service order request activity for the same service point. |
| Validation Error | If the business object fails any of the validations in the Pending state, it enters this state.<br>Activities in this state can be corrected and retried. |



Copyright © 2026, Oracle and/or its affiliates.

| State | Description |
|---|---|
| Discarded | Activities discarded in other states enter this state.<br>Enter algorithms perform the following:<br>Validate that non-final child activities can be discarded without the need for a cancel activity<br>Cancel non-final child activities<br>Send a failure notification to the requesting system |
| Waiting for Effective Date | If an orchestration activity has a future effective date, it remains in this state until the effective date is reached.<br>A Monitor algorithm transitions the activity to the next state when the activity's effective date time is reached (process date time >= effective date time). |
| Are SP and Device Ready? | Each type of orchestration activity business object has a unique set of Enter algorithms that perform operations as appropriate for the type of service order request.<br>See Service Order Activity Algorithm Types for more information about these algorithms. |
| Activity in Progress | Orchestration activities remain in this state while their child activities are processed.<br>A Monitor algorithm transitions the activity to the "Are the SP and Device Ready?" state if there are no non-final child activities related to the current activity.<br>A Monitor algorithm validates that the orchestration activity has not been in its current state for too long, based on the Expiration Days parameter on the orchestration activity's type and the Expiration Date/Time on the orchestration activity<br>An Exit algorithm resets the Expiration Date/Time on the orchestration activity such that each time the activity exits this state its Expiration Date/Time is updated. |
| Activity Error | If one or more child activities enters an Error state, the orchestration activity enters this state.<br>Activities in this state can be corrected and retried. |
| Retry | When an orchestration activity is retried after correction of an error condition, it enters this state.<br>Enter algorithms perform the following:<br>Check to determine if there are child field activities in progress that have outbound communications awaiting a response.<br>Transition any non-final child activity to the "Reject" state (the state defined as "Reject" on the child activity business object lifecycle. This is most often the "Discarded" state). |
| Completed | Orchestration activities enter this state when all child activities have successfully completed.<br>An Enter algorithm send a success notification to the requesting system. |

Use the Business Object and Algorithm portals to view additional details about this business object and its lifecycle algorithms.

### Cancel / Update Orchestration Lifecycle

The Cancel Orchestration (D1-CancelOrchestration) and Update Orchestration (D1-UpdateOrchestration) business objects have a similar lifecycle, with the following exceptions:


Copyright © 2026, Oracle and/or its affiliates.

- There is no "Waiting for Effective Date" state.
- In place of the "Are SP and Device Ready?" state, they have "Cancel Specific Activity" / "Update Specific Activity" states. Enter algorithms on these states attempt to cancel or update a specific child activity.
- In the place of "Activity In Progress" and "Activity Error" states, they have "Communication in Progress" and "Communication in Error" states.

Use the Business Object and Algorithm portals to view additional details about this business object and its lifecycle algorithms.

#### Throttling Commands Issued from Service Order Activities

Service order management can be configured to limit, or "throttle", the number of smart meter command activities initiated by a service order activity that are executed within a specified time range. For example, in a situation in which a utility initiates dozens (or hundreds) of cut for non-payment service activities on a single day, this feature allows the execution of the related remote disconnect commands to be spread out over a number of hours rather than executing all of those commands in the first hour of the day.

> **Note**
>
> Throttling is only supported for Cut Service for Non-Payment service order activities and Remote Disconnect smart meter command activities.

Configuring throttling of command processing is done via parameters on the service order activity type and the related smart meter command activity type.

##### Service Order Activity Type

The following **Business Hours** parameters on the Cut Service for Non-Payment activity type are used to define the business days and hours during which remote disconnect commands can be executed:

- **Ensure Business Hours**: Indicates if commands should be processed during specified business hours as defined by the **Work Calendar**, **Start Hour**, and **End Hour** parameters.
- **Work Calendar**: The work calendar used to determine work days on which commands can be processed.
- **Start Hour**: The start hour of the day during which commands can be processed.
- **End Hour**: The end hour of the day during which commands can be processed.

Refer to the **Business Hours** embedded help on the Cut Service for Non-Payment activity type for more details.

##### Smart Meter Command Activity Type

The **Throttle Parameters** on the Remote Disconnect activity type are used to define if throttling is enabled, and if so, the number of commands that can be processed during specific time ranges during each business day. These parameters include:

- **Throttle Option**: Specifies if throttling is allowed. If set to "Allowed", one or more time ranges must be defined in the **Schedule** list.
- **Schedule**: Defines one or more time ranges (defined by the **From** and **To** columns) and the number of records that can be processed during each (defined by the **Number of Requests** column).


Copyright © 2026, Oracle and/or its affiliates.

### Batch Processing

Processing of commands configured for throttling is performed using the Command Request Wait Monitor with Throttle Option batch control (D1-CRWTO).

This batch control monitors command activities in the "Wait For Effective Date/Time" state, and if the Throttle Option parameter (enableThrottle) is set to "Y", it will only process up to the number of records specified in the current time range defined in the **Schedule** section on the command activity type. For example, if there are 50 Remote Disconnect commands in the "Wait For Effective Date/Time" state and this batch control (with **Throttle Option** set to "Y") is run during a time range for which the **Number of Requests** is set to 30, only 30 of the commands will be transitioned to the Connection Ready state.

This batch control can be used for all commands, with the **Throttle Option** parameter set as appropriate based on the type of command activities. As of this release, throttling is only available with Remote Disconnect smart meter command activities.

Notes:

- This batch control can be used only with command activity business objects whose "Wait For Effective Date/Time" state does NOT have a monitor process defined.
- If this batch control is run more than once during a specific time range, it will process up to the number of records specified for that time range during each run. In the previous example, if the batch control were to be run twice during the same time range, the first run would process 30 records, and the second would process the remaining 20 records.

See the **Detailed Description** of this batch control for more information.

## Service Order Activity Algorithm Types

When an orchestration activity enters the "Are SP and Device Ready?" state, a set of Enter algorithms are used to evaluate the state of the service point / meter / item to determine which actions are required to complete the service request. These algorithms are based on the following algorithm types.

### Customer-Device Compatibility Check (D1-DVCOMCHK)

Algorithms of this type execute the "Customer-Device Compatibility Algorithm" defined on the orchestration activity's activity type. Algorithms of this type uses the following parameters:

- **Activity BO To Create If Compatibility Detected**: Specifies the activity business object to instantiate if the algorithm detects an in compatibility between the customer/service point and device.

Note: The base package does not include algorithm types for the "Activity Type - Customer Device Compatibility" algorithm entity.

### Connect Only If Previously Connected (D1-CONPRECON)

Algorithms of this type check if the "Connect New Device" flag has a value. If the flag is not populated, the algorithm sets the value of the flag based on the connection status of the device prior to the meter exchange (used only with Meter Exchange requests).

For additional information, refer to the D1-CONPRECON Algorithm Type.

### Create Meter Exchange Field Activity (D1-CREMTREXC)

Algorithms of this type create a service order field activity based on details provided in the algorithm's parameters. Algorithms of this type uses the following parameters:


Copyright © 2026, Oracle and/or its affiliates.

- **Activity and Specific Field Task to Create**: Specifies the type of activity business object and field task type to create for meter exchange service order field activities, as defined by the following mnemonics:

| Mnemonic | Description |
|---|---|
| activityBOToCreate | Specifies the activity business object to create. |
| specificFieldTask | Specifies the Field Task Type when creating a service order field activity. This value comes from the Field Task Type extendable lookup. |

For example, to create a service order field activity based on the D1-FieldActivity business object and the Exchange Meter field task type, these mnemonics would be configured as follows:

```
activityBOToCreate=D1-FieldActivity specificFieldTask=D1-ExchangeMeter
```

For additional information, refer to the D1-CREMTREXC Algorithm Type.

#### Decommission Removed Meter (D1-DCRMMTR)

Algorithms of this type create a decommissioning command for a removed meter (used only with Meter Exchange requests). Algorithms of this type use the following parameters:

- **Decommission Activity BO to be created**: Specifies the type of activity business object to create when decommissioning a meter. The specific activity is created, as defined by the following mnemonics:

| Mnemonic | Description |
|---|---|
| activityBOToCreate | Specifies the activity business object to create. |

For example, to create a smart meter command activity based on the D1-DeviceDecommission business object, this parameter would be configured as follows:

```
activityBOToCreate=D1-DeviceDecommission
```

For additional information, refer to the D1-DCRMMTR Algorithm Type.

#### Create Specific Activity (D1-CRSPACT)

Algorithms of this type determine if a specific activity needs to be created based on the state of the service point. Algorithms of this type use the following parameters to specify the conditions and activity to be created:

- **Field Activity BO**: Specifies the field activity business object to instantiate if the algorithm creates a field task type (see next parameter).
- **SP State and Activity BO to Create:** Specifies the type of activity business object to create based on the state of the service point. This parameter can be repeated up to 20 times. Instances of the parameter are evaluated one at a time and the first condition matching the state of the service point is used. Parameters should be ordered from the most restrictive condition to the least restrictive. This parameters uses the following mnemonics to indicate the state (any combination of the following) of the service point:


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description |
|---|---|
| servicePointConnected | Specifies if the service point is currently connected<br>Valid values are “true" and "false". |
| disconnectLocation | Specifies the “Disconnect Location” for the service point.<br>Valid values are “D1SR" (source) and "D1DV" (device). |
| deviceInstalledAtSP | Specifies if there is a device currently installed at the service point.<br>Valid values are “true" and "false". |
| installationEventStatusOverride | Specifies the value of the "Installation Status" option type of the Install Event's Status ("Pending", "Conn-PreComm", "ManualOff", etc.) |

Based on the unique combination of these mnemonics, a specific activity is created, as defined by the following mnemonics:

| Mnemonic | Description |
|---|---|
| activityBOToCreate | Specifies the activity business object to create (used most often to specify a command business object) |
| specificFieldTask | Specifies the field task type when creating a service order field activity. This value comes from the Field Task Type extendable lookup.<br>Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the field activity business object to create. |
| spTypeCategory | Specifies a service point type category. Valid values in include "D1MT" (meter), "D1IT" (item), "D1MI" (multi-item), from the SP_CATEGORY_FLG lookup.<br>Specifying this mnemonic indicates that a service order field activity should be created only if the service point’s category match the one specified by this mnemonic. |
| executeOverrideAlgorithm | Specifies whether or not to execute the algorithm specified for the **Override Device/Task Algorithm** on the service order orchestration activity type.<br>This allows the activity business object to create to be dynamically determined based on an algorithm instead of the "activityBOToCreate" or "specificFieldTask" mnemonics.<br>Valid values are “true" and "false". |

For example, the following parameter configuration would create a "Connect Service Point and Install Meter" service order field activity given the following conditions:

- Service Point Connected: False
- Disconnect Location: Source
- Device Installed at Service Point: False
- Service Point Category: Meter

```
servicePointConnected=false disconnectLocation=D1SR deviceInstalledAtSP=false
specificFieldTask=D1-ConnSPAtSrceAndInstMtr spTypeCategory=D1MT
```


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Additional values for the **SP State and Activity BO to Create** parameter can be defined using the **Field Task By Service Point Condition** zone on the Activity Type portal.

For additional information, refer to the D1-CRSPACT Algorithm Type.

### Update Device (D1-UPDDVC)

Algorithms of this type determine if an activity needs to be created to update the device based on the state of the service point and device installed at the service point. Algorithms of this type use the following parameters to specify the conditions and activity to be created.

- **Error if SP Not Connected or no Device Installed (Default is Yes)**: Indicates if the algorithm should return an error if the service point is not connected or if a device is not currently installed. Valid values are "Yes" and "Con" (continue)
- **SP State and Activity BO to Create**: Specifies the type of activity business object to create based on the state of the service point. This parameter can be repeated up to 20 times. Instances of the parameter are evaluated one at a time and the first condition matching the state of the service point is used. Parameters should be ordered from the most restrictive condition to the least restrictive. This parameters uses the following mnemonic to indicate the state (any combination of the following) of the service point:

| Mnemonic | Description |
|---|---|
| installationEventStatusOverride | Specifies the value of the "Installation Status" option type of the Install Event's Status ("Pending", "Conn-PreComm", "ManualOff", etc.) |

Based on the value of this mnemonic, a specific activity is created, as defined by the following mnemonics:

| Mnemonic | Description |
|---|---|
| activityBOToCreate | Specifies the activity business object to create (used most often to specify a command business object) |
| specificFieldTask | Specifies the Field Task Type when creating a service order field activity. This value comes from the Field Task Type extendable lookup.<br>Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the field activity business object to create. |
| spTypeCategory | Specifies a service point type category. Valid values in include "D1MT" (meter), "D1IT" (item), "D1MI" (multi-item), from the SP_CATEGORY_FLG lookup.<br>Specifying this mnemonic indicates that a service order field activity should be created only if the service point's category match the one specified by this mnemonic. |


Copyright © 2026, Oracle and/or its affiliates.

| Mnemonic | Description |
|---|---|
| alternativeFieldTask | Specifies an alternative Field Task Type to use when creating a service order field activity in the event that the device does not support the command indicated by the "activityBOToCreate" mnemonic.<br>Note: If this mnemonic is specified, the "Field Activity BO" parameter should specify the field activity business object to create.<br>A value of 'skip' will continue the evaluation of the algorithm's next parameter |

For example, the following parameter configuration would create a "Turn On Meter" service order field activity given the following conditions:

- Installation Status: Manual Off
- Service Point Category: Meter

```
installEventStatusOverride=ManualOff specificFieldTask=D1-TurnOnMeter
spTypeCategory=D1MT
```

> **Note**
>
> Additional values for the **SP State and Activity BO to Create** parameter can be defined using the **Field Task By Device Condition** zone on the Activity Type portal.

Other parameters used by algorithms of this type include:

- **Field Activity BO**: Specifies the field activity business object to instantiate if the algorithm creates a field task type (see above parameter).
- **XPath of Activity Element controlling Activity creation**: Defines an element within the activity business object schema that can be used to control whether or not this algorithm should create an activity. For example, to specify that the value of the "Connect New Device" flag be used to determine whether or not the algorithm should create an activity, this parameter could be set to "connectNewDevice".
- **Element value indication that Activity creation should not proceed**: Specifies a value for the element defined in the "XPath of Activity Element controlling Activity creation" parameter that would indicate that the algorithm should not create an activity. Valid values are based on the element defined for the "XPath of Activity Element controlling Activity creation" parameter. For example, to specify that an activity should not be created if the "Connect New Device" flag is set to "Do Not Connect / Turn On", this parameter should be set to "D1NC" (from the D1_CONNECT_NEW_DEVICE_FLG lookup).

For additional information, refer to the D1-UPDDVC Algorithm Type.

#### Remote Turn Off Turn On (D1-REMONOFF)

Algorithms of this type remotely turn a device off and on for a Back to Back service request. Algorithms of this type use the following parameters:

- **Device Incompatibility Detected Activity BO**: Specifies the activity business object the algorithm will look for. If the algorithm finds an activity of this business object, the algorithm terminates.


Copyright © 2026, Oracle and/or its affiliates.

- **Remote Connect BO**: Specifies the activity business object to instantiate when creating a remote connect command.
- **Remote Disconnect BO**: Specifies the activity business object to instantiate when creating a remote disconnect command.
- **Installation Event Status Override for Connect**: The override status to which the Installation Event Status is set after performing a remote connect command.
- **Installation Event Status Override for Disconnect**: The override status to which the Installation Event Status is set after performing a remote disconnect command.

For additional information, refer to the D1-REMONOFF Algorithm Type.

#### Check for Measurement (D1-CHKMSMT)

Algorithms of this type determine if measurements exist on activity's service point as of the service date/time. If no measurement is found, algorithms of this type create an activity to either obtain or wait for a measurement. The specific type of activity is based on the type and configuration of the device and service point. Algorithms of this type use the following parameters:

- **Activity BO To Wait For Measurement:** Specifies the activity business object to instantiate when the algorithm logic indicates it should wait for a measurement for the service point.
- **Activity BO For Field Read**: Specifies the field activity business object to instantiate when the algorithm logic indicates it should request a meter reading from the field.
- **Specific Field Task**: Specifies the field task type when creating a service order field activity for a meter reading from the field.
- **Activity BO To Wait For Scheduled Read**: Specifies the activity business object to instantiate when the algorithm logic indicates it should wait for a scheduled read for the service point.
- **Activity BO For On Demand Read - Scalar**: Specifies the activity business object to instantiate when the algorithm logic indicates it should issue an on-demand read (scalar) smart meter command.
- **Start Range for Normal Measurement Condition**: The start of the range of conditions that indicate "normal" measurements when the algorithm is searching for measurements for the service point.
- **End Range for Normal Measurement Condition**: The end of the range of conditions that indicate "normal" measurements when the algorithm is searching for measurements for the service point.
- **Minimum Range for bottom Measurement condition**: The minimum measurement condition used when searching for measurements for the service point. Used only when no measurements are found in the "normal" range defined by the "Start/End Range Normal Measurement Condition” parameters.

The following parameters on the orchestration activity type are also used by algorithms of this type when searching for measurements for the service point:

- **Look for Measurement within the Day**: Limits the search to the reference date (the service date).
- **Minimum and Maximum Offset Number of Days**: Numbers of days added to /subtracted from the reference date to expand the search period.

For additional information, refer to the D1-CHKMSMT Algorithm Type.


Copyright © 2026, Oracle and/or its affiliates.

### Algorithm Types and Orchestration Activity Business Objects

Each of the orchestration activity business objects uses a different set of these algorithm types. The table below lists which of these algorithm types are defined for each of the service order orchestration activity business objects.

| | Enable Service | Disable Service | Cut for Non-Payment | Reconnect for Payment | Meter Exchange | Back-to-Back |
|---|---|---|---|---|---|---|
| Customer-Device Compatibility Check | X | | | | | X |
| Connect Only If Previously Connected | | | | | X | |
| Create Specific Activity | X | | | | | |
| Create Meter Exchange Field Activity | | | | | X | |
| Update Device | X | X | X | X | X | X |
| Remote Turn Off Turn On | | | | | | X |
| Decommission Removed Meter | | | | | X | |
| Check For Measurement: | X | X | X | | | X |

### Cancel / Update Orchestration - Algorithm Types

Enter algorithms on the "Cancel Specific Activity" and "Update Specific Activity" states attempt to cancel or update a specific child activity. These algorithms are based on the following algorithm types.

- **Cancel Specific Activity**: Algorithms of this type cancel the specific activity (either a service order field activity or a smart meter command) that is associated to the Cancel or Update orchestration activity, based on the current status of the specific activity.
- **Update Specific Activity**: Algorithms of this type update the specific activity (either a service order field activity or a smart meter command) that is associated to the Cancel or Update orchestration activity, based on the current status of the specific activity.

| Algorithm Type | Algorithm(s) |
|---|---|
| Cancel Specific Activity (D1-CANSPACT) | Cancel Specific Activity (D1-CANSPACT) |
| Update Specific Activity (D1-UPDSPAC) | Update Specific Activity (D1-UPDSPAC) |

Use the Algorithm Type and Algorithm portals to view additional details about these algorithms.

## Understanding Service Order Activity Types

Service order activity types must be configured for each type of service order activity.


Copyright © 2026, Oracle and/or its affiliates.

Service order activity types are assigned to the following Activity Type Categories:

| Service Order Activity | Activity Type Category |
|---|---|
| Back-to-Back Service | Request Orchestration |
| Cancel Orchestration | Orchestration Maintenance |
| Cut Service for Non-Payment | Request Orchestration |
| Disable Service | Request Orchestration |
| Enable Service | Request Orchestration |
| Exchange Meter | Request Orchestration |
| Reconnect Service for Payment | Request Orchestration |
| Update Orchestration | Orchestration Maintenance |

Refer to Understanding Activity Types for more information about activity types.

### Service Order Activity Types and Divisions

Service order activity types can specify a Division, to represent the specific operating company associated with activities of that type. If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", users can only access service order activities or service order field activities with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the**Miscellaneous** tab of the **User** portal.

See Understanding Divisions for more information about divisions.

## Configuring Service Order Activity Types

You use the **SOM Activity Type** portal is to view and maintain service order activity types.

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **SOM Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured service order activity types as well as service order activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List:** This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type:** This zone provides information about the selected Activity Type
- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.


Copyright © 2026, Oracle and/or its affiliates.

- **Activity Type**: This zone provides information about the selected Activity Type
- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the “Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

### Service Order Management Configuration

The following table outlines the activity types that must be configured for the service order activity types supported by Service Order Management:

| Service Order Activity | Activity Type (Business Object) |
|---|---|
| Enable Service | Enable Service (D1–EnableServiceType) |
| Disable Service | Disable Service (D1–DisableServiceType) |
| Cut Service for Non-Payment | Cut Service for Non-Payment (D1-CutServiceForNonPaymentType) |
| Reconnect Service for Payment | Reconnect for Payment (D1–ReconnectForPaymentType) |
| Meter Exchange | Exchange Meter (D1–ExchangeMeterType) |
| Back-to-Back Service | Back-to-Back Service (D1–BackToBackServiceType) |
| Cancel Orchestration | Cancel Orchestration (D1–CancelOrchestrationType) |
| Update Orchestration | Update Orchestration (D1–UpdateOrchestrationType) |

The demonstration database contains examples of each of these service order activity types.

# Service Order Field Activities


Copyright © 2026, Oracle and/or its affiliates.

- Understanding Service Order Field Activities
- Understanding Service Order Field Activity Types
- Configuring Service Order Field Activity Types
- Configuring Service Order Field Activity Reschedule Reasons

# Understanding Service Order Field Activities

This section describes service order field activities and how they communicate with field work management systems.

Service order field activities are activities that involve sending workers into the field to perform service. This can include meter installation, meter replacement, and other activities.

Service order field activities send messages to a field work system, which in turn assigns them to crews to be completed in the field.

**Field Activity Information**

All service order field activities are based on the Field Activity (D1-FieldActivity) business object, and include the following user-accessible information:

- **Status**: The current status of the service order field activity.
- **Service Date/Time**: The date and time the service order field activity was created.
- **Service Point**: The service point associated with the service order field activity.
- **Field Task Type**: The field task type for the service order field activity. This defines the type of task and other processing details regarding how Service Order Management processes the service order field activity. See Field Task Types for more details about field task types.
- **Recipient**: The field work system service provider to which the service order field activity is sent for scheduling and assignment.
- **Device ID**: The device related to the service order field activity (if applicable).
- **Request Information**: Details of the service order request, including requester and external system information.
- **Contact Details (or Customer Information)**: Contact details for the customer associated with the service order request.
- **Address Information**: The address of the service point associated with the service order field activity.

The field activity business object also contains other information that is populated by algorithms and scripts as the service order field activity is processed by the system.

- How Do Service Order Field Activities Work?
- Service Order Field Activity Processing
- Service Order Field Activity Communication
- Unrelated Pickup Orders

## How Do Service Order Field Activities Work?

At a high level, service order field activities work as follows:


Copyright © 2026, Oracle and/or its affiliates.

### Create Field Activity

A service order orchestration activity creates a service order field activity based on the current state of the service point/meter/item.

### Retrieve Required Data

The service order field activity uses a set of pre-processing algorithms to derive and populate data needed by the activity, such as the device, service point, address, effective date, and others.

### Request Appointment (Optional)

If the service order field activity task type specifies that field tasks of this type require an appointment, the service order field activity checks for available appointment slots in the field work system and sends a notification to the appointment handling system.

### Create Outbound Communication

The service order field activity creates an outbound communication to send the service order field activity to the field work system. The outbound communication gathers the information required by the field work system before being sent. This information is retrieved by a set of processing scripts defined on the field task type.

### Receive Inbound Communication

When the service order field activity has been completed, the field work system sends an inbound communication back to Service Order Management.

Inbound communications can contain Field Activity Remarks (entered by field resources when they perform and complete their field work. If the Field Activity Remarks reference completion events, they are executed.

The inbound communications create completion events as defined on the field task type. If the service order field activity was successfully completed, it creates the "Completion Events When Successful" completion events. If the service order field activity was canceled, it creates the "Completion Events When Canceled" completion events.

### Execute Completion Events

After receiving the inbound communication, a service order field activity algorithm transitions any active completion events into their executed state.

### Complete Processing

The service order field activity completes its processing by doing the following:

- Updating the parent orchestration activity
- Sending a success response to the requester
- Transitioning the parent orchestration activity to the next state in its lifecycle
- Sending a service order field activity completion outbound communication to subscribing systems.

## Service Order Field Activity Processing

This section outlines how service order field activities are processed.


Copyright © 2026, Oracle and/or its affiliates.

### Pre-Processing, Validation, and Post-Processing Algorithms

When service order field activities are first instantiated, a set of pre-processing algorithms populate and derive information needed for the activity, such as the activity type, service point, device, address, effective date, and other information.

Validation algorithms validate this information when first retrieved and when updated.

When service order field activities are completed, a post-processing algorithm populates the activity end date/time.

### Service Order Field Activity Lifecycle

As a service order field activity moves through its lifecycle, it triggers various business processes based on the type of service order field activity. The table below outlines the lifecycle for the Field Activity (D1-FieldActivity) business object.

| State | Description |
|---|---|
| Pending | The initial state for service order field activities.<br>An Enter algorithm sends an acknowledgement to the requesting system.<br>The activity is transitioned to the next state via a monitor process. |
| Validate | Enter algorithms perform the following:<br>Validate Activity Type (and transition to error if invalid)<br>Derive and validate service order field activity recipient<br>Validate duplicate and conflict service order field activities<br>Derive and validate service order field activity service point<br>Derive and validate service order field activity device<br>Validate address constituents<br>Check for any existing cut service restrictions<br>The activity is transitioned to the next state via a monitor process. |
| Validation Error | If the business object fails any of the validations in the Pending state, it enters this state.<br>Enter algorithms perform the following:<br>Create a To Do based on specified To Do Type and To Do Role<br>Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed.<br>Activities in this state can be corrected and retried. |
| Waiting to Request | If a service order field activity has a future effective date, it remains in this state until the effective date is reached.<br>A Monitor algorithm transitions the activity to the next state when the activity's effective date time is reached (process date time >= effective date time).<br>An Enter algorithm sets the "Allow Child to Transition Parent Activity Based On Effective Date" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed. |


Copyright © 2026, Oracle and/or its affiliates.

| State | Description |
| --- | --- |
| Waiting for Appointment | If the service order field activity passes its validations and the effective date has been reached, the activity enters this state.<br>Enter algorithms perform the following:<br>Evaluate if an appointment is required for field tasks of this type. If not, the activity transitions to the "Communication in Progress" state.<br>Create a To Do if an appointment is necessary but the system is not able to send an appointment request.<br>Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed.<br>Send a notification to the appointment handling system.<br>Monitor algorithms perform the following:<br>Verify if an appointment has been supplied.<br>Send a notification to the appointment handling system.<br>The activity is transitioned to the next state via a monitor process.<br>See Waiting for Appointment for more information about this state. |
| Communication in Progress | Service order field activities enter this state following the "Waiting for Appointment" or "Retry" states.<br>Enter algorithms perform the following:<br>Create an outbound communication for the service order field activity (see Communication in Progress for more information).<br>Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed.<br>Monitor algorithms perform the following:<br>Check for existing child communications.<br>Check that the activity hasn’t timed out. |
| Discarded | Activities discarded in other states enter this state.<br>Enter algorithms perform the following:<br>Cancel outstanding outbound communications.<br>Cancel outstanding completion events.<br>Populate the cancel reason.<br>Send a failure notification to the requesting system.<br>Transition the parent activity to the "Activity Error" state (see Service Order Orchestration Activity Lifecycle for more information).<br>Check if a Cancel Orchestration activity is required. |


Copyright © 2026, Oracle and/or its affiliates.

| State | Description |
| --- | --- |
| Communication Error | If an outbound or inbound communication an Error state, the service order field activity enters this state.<br>Monitor algorithms perform the following:<br>Check that the activity hasn’t timed out.<br>Enter algorithms perform the following:<br>Create a To Do based on specified To Do Type and To Do Role.<br>Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed.<br>Activities in this state can be corrected and retried. |
| Retry | When a service order field activity is retried after correction of an error condition, it enters this state.<br>Enter algorithms perform the following:<br>Check to determine if there are associated outbound communications in progress.<br>Cancel any outstanding outbound communications. |
| Execute Completion Events | After an inbound communication is received, it enters this state.<br>Enter algorithms perform the following:<br>Executes completion events defined on the field task type (these completion events were initially created by the inbound communication).<br>Evaluates the "Field Activity Completed" flag on the service order field activity. If this is set to "No", the service order field activity is transitioned to the "Canceled In Field" state.<br>The activity is transitioned to the next state via a monitor process.<br>See Execute Completion Events for more information about this state. |
| Completion Event Error | If an error occurs during completion event processing, the service order field activity enters this state.<br>Monitor algorithms perform the following:<br>Check that the activity hasn’t timed out.<br>Enter algorithms perform the following:<br>Create a To Do based on specified To Do Type and To Do Role.<br>Set the "Allow Child to Transition Parent Activity" flag to “yes”. This allows the service order field activity to transition the parent orchestration activity if needed.<br>Activities in this state can be corrected and retried. |
| Completed | Service order field activities enter this state when all completion events have successfully completed.<br>Enter algorithms perform the following:<br>Update the parent orchestration activity.<br>Send a success response to the requester.<br>Transition the parent orchestration activity to the next state in its lifecycle.<br>Send a service order field activity completion outbound communication to subscribing systems. |


Copyright © 2026, Oracle and/or its affiliates.

| State | Description |
| --- | --- |
| Canceled in Field | If the "Field Activity Completed" flag on the field activity is set to "No", the service order field activity enter this state.<br>Enter algorithms perform the following:<br>Send a failed response to the requester.<br>Transition the parent orchestration activity to the "Activity Error" state.<br>Create a To Do to notify users that the service order field activity has been canceled. |

### Waiting for Appointment

When a service order field activity enters the "Wait for Appointment" state, it first determines if an appointment is necessary for the service order field activity. If not, the activity moves on to the "Communication in Progress" state (see below).

If an appointment request cannot be sent for some reason, the service order field activity creates a To Do item to alert a user to attempt to manually request an appointment. Otherwise, the service order field activity sends an outbound message to the field work system requesting an appointment. based on the appropriate processing role defined on the."Send Notification to Appointment Handling System - Enter" algorithm.

| Processing Role | Outbound Communication Business Object |
| --- | --- |
| Appointment Response (default)<br>Used if:<br>An appointment is required and needs to be scheduled<br>Appointment has been set | Send Appointment Response Outbound Message (D1-SendApptRespOutboundMsg)<br>Note: An outbound message must be created based on this business object. |

The response from the field work system can be received by creating an Inbound Web Service that references the "Book selected appointment to Field Activity" (D1-BookAppt) service script.

While in this state, monitor algorithms verify if an appointment has been supplied and send notifications to the field work system.

### Communication in Progress

Service order field activity communications are records of messages sent between Service Order Management and an external field work system. Communications can flow both outbound and inbound.

See Service Order Field Activity Communication for more information about service order field activity communication.

### Manually Completing Service Order Field Activities

Service order field activities in the "Communication in Progress" state can be manually completed or canceled by clicking **Complete** in the **Record Actions** section of the **Activity** zone. This opens the **Field Activity Manual Completion** page where the user can enter appropriate details to manually complete the field activity (such as completion information, service point information, device information, and so on). When submitted, this page creates an inbound communication and transitions the field activity to the next state as appropriate.


Copyright © 2026, Oracle and/or its affiliates.

Note: Implementations can define custom service scripts for each field task type in the **Manual FA Completion Controls** section on the Field Task Type extendable lookup to customize the sections displayed on the **Field Activity Manual Completion** page.

#### Execute Completion Events

After receiving the inbound communication, the service order field activity enters the "Execute Completion Events" state.

The inbound communication will have previously created completion events for the service order field activity, based on those defined on the field task type or those referenced by field activity remarks. These creation events begin in the "Pending" state.

An Enter algorithm transitions completion events associated with the service order field activity into their "Executed" state.

## Service Order Field Activity Communication

This section outlines how service order field activities communicate with field work systems.

When a service order field activity enters the "Communication in Progress" state, it sends an outbound communication to the field work system, and waits for an inbound communication response.

See **Understanding the Service Order Field Activity Communication Process** below for more information about the role of communications in the service order field activity communication process.

#### Outbound Communications

Outbound Communications represent messages sent from Service Order Management to. an external field work system Outbound communications use the following types of objects:

#### Outbound Communication Business Objects

An outbound communication business object exists for each type of message to be sent to an external system. For service order field activities, the following base package outbound communication objects can be used.

| Type of Outbound Communication | Outbound Communication Business Object |
|---|---|
| Initial service order field activity outbound communication | Field Activity Outbound Communication (D1-FieldActivityOBComm) |
| Modify outbound communication<br>Used to send an update to a service order field activity previously sent to the field work system. | Field Activity Outbound Communication (D1-ActivityModifyOBComm) |

#### Outbound Message Types

A outbound message type must also be created for each type of message to be sent to an external system. Again, this is based on the types of messages the system is designed to accept. For service order field activities, the following outbound message types are needed:


Copyright © 2026, Oracle and/or its affiliates.

| Type of Outbound Communication | Outbound Message Type |
|---|---|
| Initial Service Order Field Activity Message | Field Activity Outbound Message |
| Modify Existing Service Order Field Activity | Modify Field Activity Outbound Message |

Refer to the Oracle Utilities Application Framework documentation for more information about outbound message types.

### External Systems

You must also create an External System for each external system to which Service Order Management will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch, XAI, or Real-time)
- Message Sender (if Processing Method is set to Real-time or XAI)
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

To continue the example above, you might create the following external system:

| External Application | | |
|---|---|---|
| **Outbound Message Type** | **Processing Method** | **Batch Control** |
| Field Activity Outbound Message | Batch | Sync Request Monitor (F1-SYNRQ) |
| Modify Field Activity Outbound Message | Batch | Sync Request Monitor (F1-SYNRQ) |

Refer to the Oracle Utilities Application Framework documentation for more information about external systems.

### Inbound Communications

Inbound Communications represent messages sent from an external field work system to Service Order Management. Inbound communications are typically sent in response to a service order field activity. Inbound communications use the following types of objects:

### Inbound Communication Business Objects

An inbound communication business object must be created for each type of message to be received from an external system. For service order field activities, the following base package inbound communication object can be used.

| Inbound Communication Business Object |
|---|
| Field Activity Inbound Communication (D1-FieldActivityIBComm) |

### Inbound Web Service


Copyright © 2026, Oracle and/or its affiliates.

You must also create an Inbound Web Service for each type of message to be received from an external system. Inbound web services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of inbound web services you need to create is based on the types of messages the system is designed to send. To continue the example above, you might create the following inbound web services:

| Inbound Web Service | Schema (Inbound Communication Business Object) |
|---|---|
| Field Activity Inbound Communication | Field Activity Inbound Communication D1-FieldActivityIBComm |

Refer to the Oracle Utilities Application Framework documentation for more information about Inbound Web Services.

### Field Activity Remarks

Inbound communications can contain activity remarks, which represent notes entered by the field worker as they perform and complete their field work. These can be solely informational, or can reference completion events via the "Remark Processing" section of the Field Activity Remark Type extendable lookup. This allows information sent with the inbound communication to initiate business processing if necessary.

Completion events specified on this extendable lookup are created by the inbound communication, and then executed when the service order field activity enters the "Execute Completion Events" state.

#### Understanding the Service Order Field Activity Communication Process

This section provides an overview of the communication process that takes place when a service order field activity is initiated. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific base package objects used by Service Order Management

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

| Step | Process | Base Package Objects |
|---|---|---|
| 1. | An orchestration activity creates a service order field activity as part of its processing.<br>A service order field activity business object is instantiated for the command. | Field Activity Business Object: Field Activity (D1-FieldActivity) |
| 2. | When the service order field activity enters the Communication in Progress tate, it creates an outbound communication. | Outbound Communication Business Object: Field Activity Outbound Communication<br>(D1-FieldActivityOBComm) |
| 3. | A Enter algorithm on the "Awaiting Response" state of the outbound communication retrieves information needed by the outbound message to be sent to the field work system based on processing scripts specified on the field task type. | Enter Algorithm: Populate Send Detail for Field Activity (D1-POPSNDDTL) |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Process | Base Package Objects |
|---|---|---|
| 4. | A Enter algorithm on the "Awaiting Response" state of the outbound communication creates an outbound message. | Enter Algorithm: Create Outbound Message (D1-COUTMSG)<br>Note: An outbound message type for this message is not included in the base package. |
| 5. | The outbound message is sent to middleware components via an External System and Batch Control.<br>Middleware components utilize Business Process Execution Language (BPEL). | External System: MWM<br>Batch Control: Sync Request Monitor<br>(F1-SYNRQ) |
| 6. | The middleware converts the outbound message from SOM format into the format used by the field work system, and sends the message to the field work system. | |
| 7. | When the field work system sends a response, the middleware receives the response message from the field work system, and converts it from the format used by the field work system to SOM format and invokes an Inbound Web Service. | Inbound Web Service: D1-FieldActivityIBComm |
| 8. | The Inbound Web Service picks up the message, and creates a corresponding inbound communication.<br>The specific type of inbound communication business object created is determined by the Inbound Web Service. | Inbound Web Service: D1-FieldActivityIBComm<br>Inbound Communication Business Object: Field Activity Inbound Communication<br>(D1-FieldActivityIBComm) |
| 9. | The inbound communication identifies the parent outbound communication. | Outbound Communication Business Object: Field Activity Outbound Communication<br>(D1-FieldActivityOBComm) |
| 10. | The inbound communication creates the completion events defined on the field activity field task type (Successful or Canceled, as appropriate) in the "Pending" state.<br>If the inbound communication contains field activity remarks, it also executes any field activity remark completion events. | Inbound Communication Business Object: Field Activity Inbound Communication<br>D1-FieldActivityIBComm |
| 11. | The inbound communication updates the outbound communication.<br>This update is performed by an Enter algorithm on the “Completed” Status of the inbound communication business object’s lifecycle. | Inbound Communication Business Object: Field Activity Inbound Communication<br>D1-FieldActivityIBComm<br>Outbound Communication Business Object: Field Activity Outbound Communication<br>(D1-FieldActivityOBComm) |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Process | Base Package Objects |
|---|---|---|
| 12. | The outbound communication updates the “Completion Flag” and the original service order field activity business object.<br>This update is performed by an Enter algorithm on the “Completed” Status of the outbound communication business object’s lifecycle. | Outbound Communication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect)<br>Field Activity Business Object: Field Activity (D1-FieldActivity) |

# Unrelated Pickup Orders

When field work crews are out performing field work, it’s possible that they will encounter other work unrelated to their current task that needs to be done. This type of work can be as simple as trimming a tree whose branches are too close to power lines, or the replacement of a meter for a different customer or service point. These types of task are referred to as "unrelated pickup activities." Crews can either work the field activity or leave it to be assigned to another crew at a later date.

When the crew creates an unrelated pickup activity in the field work system, it is sent to Service Order Management, and a corresponding service order field activity is created in the system.

Unrelated pickup activities can be created via one of the following Inbound Web Services:

- Field Activity Asynchronous Req Inbound (D1-FARequestAsynchronous)
- Field Activity Synchronous Req Inbound (D1-FARequestSynchronous)

Once created, the are processed like any other service order field activity. If the pickup activity was completed in the field before being sent to Service Order Management, it will quickly move through its lifecycle (as now further action is needed) until it reaches the "Completed" state.

### Retrieving Service Point Information

If the unrelated pickup activity is customer-related it will require service point information to be created. This information can be queried by the field work crew via the "Field Work Service Point Query" (D1-FieldWorkSPQuery) Inbound web service.

This service uses set of service point criteria to allow the field crew to search for a service based upon either service point or device information. The service returns a list of service points that is configurable in length. If the number of results is larger than the configured maximum length the service indicates that additional records exist and the crew can request another set of results allowing them to identify the proper service point to associate to the activity.

There are times when an unrelated pick-up activity is identified but the field crew is out-of-coverage (i.e. no network connection) and they will not be able to immediately verify service point information. In this type of situation, the crew can input the service point criteria fields and create the activity, which, when imported into Service Order Management, will attempt to identify the service point based upon the information provided. If the service point can be uniquely identified everything should operate as normal. If the service point cannot be uniquely identified then the service order field activity is set to the error state.


Copyright © 2026, Oracle and/or its affiliates.

# Understanding Service Order Field Activity Types

A single service order field activity type must be configured to support communication with external field work systems such as Oracle Utilities Module Workforce Management. Service order activity types are assigned to the “Field Activity” Activity Type Category.

Refer to Understanding Activity Types for more information about activity types.

### Service Order Field Activity Types and Divisions

Service order field activity types can specify a Division, to represent the specific operating company associated with activities of that type. If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", users can only access service order activities or service order field activities with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the**Miscellaneous** tab of the **User** portal.

See Understanding Divisions for more information about divisions.

# Configuring Service Order Field Activity Types

You use the **SOM Activity Type** portal is to view and maintain service order field activity types.

Refer to Understanding Activity Types for more information.

You can access the portal by selecting **Admin**, **Communication**, **SOM Activity Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", this zone displays only unsecured service order field activity types as well as service order field activity types with a Division (or Divisions) associated with the user on the **User Miscellaneous Information** zone on the **Miscellaneous** tab of the **User** portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Activity Type List:** This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type:** This zone provides information about the selected Activity Type
- **Activity Type List**: This zone works differently than the typical zone that list types in that it displays both those activity types that have been configured as well as those activity types that have yet to be configured. Broadcast a record to display the details of the selected record.
- **Activity Type**: This zone provides information about the selected Activity Type
- **Field Task By Device Condition**: This zone is used to configure field tasks such as Turn On Device or Turn Off Device, based on the device condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the Update Device (D1-UPDDVC) algorithm type. This allows the **SP State and**


Copyright © 2026, Oracle and/or its affiliates.

**Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes" , only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

- **Field Task By Service Point Condition**: This zone is used to configure field tasks such as “Install Device” or “Connect SP”, based the service point condition. The configurations defined in this zone are used by the **SP State and Activity BO to Create** parameter for algorithms based on the “Create Specific Activity (D1-CRSPACT) algorithm type. This allows the **SP State and Activity BO to Create** parameter to be defined more than the 20 times allowed on the algorithm. Note that configurations defined in this zone can also include **Service Point Type**.

> **Note**
>
> If the **Restrict Division** setting in the **Control by Division** section in the MDM Master Configuration is set to "Yes", only service point types available to the user (based on division), and only field task types considered valid for the service point type can be defined in this zone.

### Service Order Management Configuration

The following table outlines the activity types that must be configured for the service order field activity type supported by Service Order Management:

| Field Activity | Activity Type (Business Object) |
|---|---|
| Field Activity | Field Activity (D1–FieldActivityType) |

The demonstration database contains examples of each of these service order activity types.

# Configuring Service Order Field Activity Reschedule Reasons

The Reschedule Reason (D1-RescheduleReasonLookup) extendable lookup can be used to define reasons for rescheduling field activities. Reschedule reasons are captured on field activities, as well as their parent service order activity and an update orchestration activity.

Use the following steps to define service order field activity reschedule reasons:

1. Select **Admin**, **General**, **Extendable Lookup** to access the **Extendable Lookup Query** portal.
2. Enter “D1-RescheduleReasonLookup” in the **Business Object** field (or enter “Reschedule Reason” in the **Description** field) and click **Search**.
3. Click the “Reschedule Reason” link in the **Description** column of the search results.
4. Click **Add**, and enter the following details for the reschedule reason ( fields not listed below are optional):


Copyright © 2026, Oracle and/or its affiliates.

- **Reschedule Reason**: A code used to designate the reschedule reason.
- **Status**: The status of the reschedule reason.
- **Description**: A description of the reschedule reason. This is what will appear in **Reschedule Reason** drop-down list on the service order field activity.

5. Click **Save**.

# Field Task Types


- Understanding Field Task Types
- Configuring Field Task Types


## Understanding Field Task Types

A service order field activity's field task type defines details about the type of task to be performed and how the system will process the activity.

### Field Task Type Information

Field task types are values for the Field Task Type (D1-FieldTaskTypeLookup) extendable lookup. Each field task type value includes the following information:

- **Routing**: Indicates if field tasks of this type can only be performed at a service point. Valid values are "SP Required" and "Pass-Through".
- **Appointment Option**: Indicates if an appointment (via a mobile workforce application) is required or applicable to field tasks of this type. Valid values are "Not Applicable", "Optional", and "Required".
- **Completion Events When Successful**: One or more completion events that are executed upon successful completion of field tasks of this type.
- **Completion Events When Canceled**: One or more completion events that are executed upon cancellation of field tasks of this type.
- **Duplicate Task Type Information**: Defines processing rules for handling potential duplicate field tasks, including:
  - **Allow Duplicates**: Specifies whether or not duplicate field tasks are allowed.
  - **Duplicate Threshold**: A number of hours used to determine if a newly instantiated field task type should be considered a duplicate.
  - **Field Task Types**: A list of one or more field task types that are considered to be duplicates of the field task type.
- **Conflict Task Type Information**: Defines processing rules for handling potentially conflicting field tasks, including:
  - **Allow Conflicts**: Specifies whether or not conflicting field tasks are allowed.
  - **Conflict Threshold**: A number of hours used to determine if a newly instantiated field task type should be considered a conflict.
  - **Field Task Types**: A list of one or more field task types that are considered to conflict with the field task type.
- **Processing Scripts**: Defines one or more processing scripts to extract supplemental information needed by the mobile workforce application to schedule field tasks of this type.



Copyright © 2026, Oracle and/or its affiliates.

- **Manual FA Completion Controls**: Defines an optional Service Script used to customize the sections displayed on the **Field Activity Manual Completion** page when manually completing service order field activities.

# Configuring Field Task Types

Field task types are configured using the extendable lookup portal.

You can access the extendable lookup portal from the **Admin > General > Extendable Lookup**.

Use the **Extendable Lookup Search** zone to search for and select the Field Task Type (D1-FieldTaskTypeLookup) extendable lookup.

The following zones may appear as part of the portal's **Main** tab page:

- **Extendable Lookup List:** This zone displays a list of values for the Field Task Type extendable lookup.
- **Extendable Lookup List:** This zone provides information about the selected value.

### Service Order Management Configuration

The following table outlines the field task types that must be configured to support each of the service order activity types supported by Service Order Management:

| Service Order Activity | Field Task Types |
|---|---|
| Enable Service — Meters | Connect SP at Source: D1-ConnectSPAtSource<br>Connect SP at Meter: D1-ConnectSPAtMeter<br>Connect SP at Source and Turn On: D1-ConnectSPAtSourceAndTurnOn<br>Connect SP at Meter and Turn On: D1-ConnectSPAtMeterAndTurnOn<br>Connect SP at Source and Install Meter: D1-ConnSPAtSrceAndInstMtr<br>Connect SP at Meter and Install Meter: D1-ConnSPAtMtrAndInstMtr<br>Install Meter: D1-InstallMeter<br>Turn On Meter: D1-TurnOnMeter |
| Enable Service — Items | Connect SP at Source: D1-ConnectSPAtSource<br>Item - Connect SP at Device: D1-ConnectSPAtDevice<br>Item - Connect SP at Source and Turn On: D1-ConnSPAtSrceAndTurnOnDvc<br>Item - Connect SP at Device and Turn On: D1-ConnectSPAtDvcAndTurnOn<br>Connect SP at Source and Install Device: D1-ConnSPAtSrceAndInstDvc<br>Connect SP at Device and Install Device: D1-ConnSPAtDvcAndInstDvc<br>Item - Install Device: D1-InstallDevice<br>Turn On Item: D1-TurnOnItem |
| Disable Service — Meters | Turn Off Meter: D1-TurnOffMeter |
| Disable Service — Items | Turn Off Item: D1-TurnOffItem |


Copyright © 2026, Oracle and/or its affiliates.

| Service Order Activity | Field Task Types |
|---|---|
| Cut Service for Non-Payment — Meters | Cut for Non-Payment: D1-CutForNonPayment |
| Cut Service for Non-Payment — Items | Item - Cut for Non-Payment: D1-CutItemForNonPayment |
| Reconnect Service for Payment — Meters | Reconnect for Payment: D1-ReconnectForPayment |
| Reconnect Service for Payment — Items | Item - Reconnect for Payment: D1-ReconnectItemForPayment |
| Meter Exchange | Exchange Meter: D1-ExchangeMeter |
| Item Exchange | Exchange Device: D1-ExchangeDevice |
| Back-to-Back — Meters | Read Meter: D1-ReadMeter |

# Service Order Management External Applications

The external systems used with Service Order Management must be defined as External Applications using the "External Application" (D1-ExternalApplication) business object. Examples of external systems can include:

- A customer information system (such as Oracle Utilities Customer Care and Billing)
- A field work system
- An asset management system (such as Oracle Utilities Operational Device Management or Oracle Utilities Work and Asset Management)

Information defined for external system service providers used by Service Order Management include:

- **Our Name/ID in Their System**: This is the value that the field work system uses to identify our system.
- **Utility Device ID Type**: This is the Device ID Type that will be used when communicating with the external application and it will be the assumed Device ID Type for any device identifiers sent from the external application.
- **Utility Service Point ID Type**: This is the Service Point ID Type that will be used when communicating with the external application and it will be the assumed Service Point ID Type for any service point identifiers sent from the external application.

Refer to Understanding External Applications and Configuring External Applications for more information about external applications.

### Processing Roles

The external application’s processing roles define how data relevant to the field work system is sent and/or created.

Field work service providers can use the following processing roles:

- **Activity Notification**: Used to send notifications to subscribing and/or requesting systems about the status of orchestration and/or service order field activities.
- **Appointment Request**: Used to send a request for an appointment to the field work system.
- **Cancelation Activity**: Used to send notifications to requesting systems when canceling orchestration and/or service order field activities.


Copyright © 2026, Oracle and/or its affiliates.

- **Collection Details**: Used to retrieve details about collections processing (used with "Cut Service for Non-Payment" and "Restore Service for Payment" orchestration activities).
- **Customer Contact**: Used to send a contact to a customer regarding a service request
- **Field Activity**: Used to send a service order field activity to the field work system.
- **Field Activity Completion**: Used to send a notification regarding completion of a service order field activity.
- **Interim Status Update**: Used to send updates regarding the status of orchestration and service order field activities to requesting systems.
- **Meter Exchange Mapping**: Used to define how to define different types of meter exchanges based specific roles and device configurations. This can provide context to field crews to help ensure they install the correct type of device and device configuration when exchanging a meter.
- **Response - Appointment**: Used to send a request for an appointment to the field work system.
- **Response - Fail**: Used to send a response to an external system when Service Order Management fails to respond.
- **Response - Missed Appointment**: Used to send a response to the field work system when notification of a missed appointment is received.
- **Response - Negative Acknowledgement**: Used to send a negative acknowledgement response to an external system in the event that a request is rejected.
- **Response - Received**: Used to send a response to an external system to acknowledge receipt of a request.
- **Response - Success**: Used to send a response to an external system when Service Order Management successfully processes a request.
- **Send Field Activity Remark**: Used to send a service order field activity remark to a subscribing system
- **Update Activity**: Used to send notifications to requesting systems when updating orchestration and/or service order field activities.

# Defining Settlement Options


- Settlement Configuration Overview
- Settlement Subscription Types
- Settlement Transaction Exception Types
- Settlement Units
- Settlement Calculations
- Settlement Item Types
- Settlement Item Quantity Types
- Market Contract Types
- Market Products
- Settlement Module Configuration




Copyright © 2026, Oracle and/or its affiliates.

# Settlement Configuration Overview

This section provides information about set up and configuration of administration data used by calculations and processes provided with Oracle Utilities Market Settlements Management.

# Settlement Subscription Types


- Understanding Settlement Subscription Types
- Configuring Settlement Subscription Types


## Understanding Settlement Subscription Types

Settlement subscription types are collections of properties defining a class of settlement subscriptions. Settlement subscription types also control valid values for various attributes of settlement subscriptions.

Settlement subscription types are defined by:

- **Subscription Business Object**: The business object used for settlement subscriptions based on settlement subscriptions of this type
- **Transaction Business Object**: The business object used for settlement transactions based on settlement subscriptions of this type
- **Default Recipient**: The recipient that will be defaulted for settlement subscriptions of this type. A settlement subscription's recipient dictates the system to which results of settlement transactions are sent.
- **Subscription Category**: Specifies the category of settlement subscriptions based on settlement subscriptions of this type, such as "Initial", "Final", and so on. Valid categories are defined in the SUBSCRIPTION_CAT_FLG lookup
- **Transaction Retention Mode**: Defines how settlement transactions based on settlement subscriptions of this type are retained. "Keep All" indicates that all settlement transactions are retained when a replacement transaction is created (older transactions are sent to a status of "Superseded"). "Keep One Record per Period" indicates that only a single settlement transaction is retained for a time period (older transactions are deleted when the new one is completed).
- **Processing Timetable Type**: A processing timetable type used when creating settlement transactions based on settlement subscriptions of this type via batch processing.
- **Valid Settlement Calculation Groups**: A list of valid settlement calculation groups for settlement subscriptions of this type
- **Valid Recipients**: A list of valid recipients for settlement subscriptions of this type
- **Communication**: The outbound communication type used to create of outbound communications when sending the results of settlement transactions based on settlement subscriptions of this type

Refer to Processing Timetable Types for more information about setting up processing timetable types.



Copyright © 2026, Oracle and/or its affiliates.

Refer to Configuring Settlement Units for more information about setting up settlement units and their respective dimension combinations, factors, and factor values.

For a deeper functional understanding, refer to the About Settlement Subscriptions or About Settlement Calculation sections of the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide*.

## Configuring Settlement Subscription Types

Refer to Understanding Settlement Subscription Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Subscription Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Subscription Type List**: This zone lists all usage subscription type records. Broadcast a record to display the details of the selected record.
- **Settlement Subscription Type**: This zone displays details for the selected usage subscription type.

The settlement subscription type defines a number of lists for valid objects that can be used in conjunction with the overall usage calculation process:

- Valid Settlement Calculation Groups
- Valid Recipients

Settlement subscription types also define the outbound communication type used when sending the results of settlement transactions based on settlement subscriptions of this type.

# Settlement Transaction Exception Types


- Understanding Settlement Transaction Exception Types
- Configuring Settlement Transaction Exception Types


## Understanding Settlement Transaction Exception Types

Settlement Transaction Exception Types define the groupings of exceptions for a settlement transaction based on their functional similarity. This provides a way to define Settlement Transaction Exceptions in a distinct enough way to understand the root issue that was generated from the usage calculation rule.

For a deeper functional understanding of Settlement Calculation, refer to the About Settlement Calculations section of the *Oracle Utilities Meter Solution Business User Guide*.

## Configuring Settlement Transaction Exception Types

This portal is used to display and maintain a Settlement Transaction Exception Type.

Refer to Understanding Settlement Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Transaction Exception Type**.



Copyright © 2026, Oracle and/or its affiliates.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Transaction Exception Type List**: displays all of the Settlement Transaction Exception Types so you can choose the one you want to display in more detail
- **Settlement Transaction Exception Type**: shows the specific configuration for the selected Settlement Transaction Exception Type

# Settlement Units


- Understanding Settlement Units
- Configuring Settlement Units


## Understanding Settlement Units

Settlement units define the “lowest common denominator set” of aggregation dimensions. For example, if aggregations are performed based on Rate Class, Strata, Procurement Group, and Supplier, the following dimensions would form the “lowest common denominator set” of these dimensions (since these are shared by different suppliers):

- Rate Class
- Strata
- Procurement Group

Settlement units define factor values based on unique combinations of dimension values. Different factor values can be defined for each unique combination of a settlement unit’s dimensions. For example, a Rate Class, Strata, and Procurement Group settlement unit could have different factor values for each unique combination of those dimensions. These multi-variable factor values can be retrieved by value derivation algorithms when performing aggregation, estimation, and forecasting calculations based on the same dimensions as defined for the settlement unit.

For example, an aggregation measuring component that aggregates data based on a specific combination of Rate Class, Strata, and Procurement Group could retrieve a “loss” factor value from a Rate Class, Strata, and Procurement Group settlement unit based on the same combination of dimensions. See Applying Losses for an example of how settlement units can be referenced in aggregation calculations.

Aggregation processing also uses settlement units to generate counts of accounts for forecasting based on combinations of these dimensions.

Settlement units are defined by:

- **Dimensions**: The attributes by which aggregated data should be analyzed
- **Dimension Combinations**: Unique combinations of the settlement unit’s dimensions (defined in the **Settlement Unit Dimension Combination** zone)
- **Valid Factors**: Valid multi-variable factors (defined in the **Settlement Unit Valid Factors** zone)

Refer to Configuring Settlement Units for more information about setting up settlement units and their respective dimension combinations, factors, and factor values.



Copyright © 2026, Oracle and/or its affiliates.

## Configuring Settlement Units

Refer to Understanding Settlement Units for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Unit**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Unit**: This zone displays details for the settlement unit, including configured dimensions.
- **Settlement Unit Dimension Combinations**: This zone displays combinations of the settlement unit's dimensions, based on the dimensions configured in the **Settlement Unit** zone. Click **Add Dimension Combination** in the zone's title bar to add a new combination.
- **Settlement Unit Valid Factors**: This zone displays valid multi-variable factors for the settlement unit. Click **Add Factor** in the zone's title bar to add a new factor. Values for these factors are defined on the **Multi-Variable Factor** tab.

The following zones appear as part of the portal's **Dimension Combinations** tab page:

- **Settlement Unit Dimension Combinations**: This zone displays a list of dimension combinations for the settlement unit. Broadcast a record to display the details of the selected record. Click **Add Dimension Combination** in the zone's title bar to add a new combination.
- **Dimension Combination Values**: This zone displays factor values for each multi-variable factor defined for the selected dimension combination. You can use this zone to add and/or edit factor values.

The following zones appear as part of the portal's **Multi-Variable Factor** tab page:

- **Multi-Variable Factor List**: This zone displays a list of multi-variable factors for the settlement unit. Broadcast a record to display the details of the selected record.
- **Multi-Variable Factor Value List**: This zone displays factor values for each dimension combination for the selected multi-variable factor. You can use this zone to add and/or edit factor values.

# Settlement Calculations

- Settlement Calculation Groups
- Settlement Calculation Rules

## Settlement Calculation Groups

- Understanding Settlement Calculation Groups
- Understanding Bulk Settlement Calculation Groups
- Configuring Settlement Calculation Groups
- Configuring Bulk Settlement Calculation Groups


Copyright © 2026, Oracle and/or its affiliates.

## Understanding Settlement Calculation Groups

Settlement calculation groups are collections of settlement calculation rules that are used to perform settlement calculations, including calculating service point quantities, totaling consumption and usage for settlement units, application of losses, and allocation of unaccounted for energy (UFE). During the settlement transaction process, the system executes the settlement calculation rules defined in the settlement calculation group referenced on the settlement subscription. The rules within a settlement calculation group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Settlement calculation groups are associated with specific settlement subscriptions and settlement subscriptions types (or both). When assigned to settlement subscriptions, settlement calculation groups contain the settlement calculation rules used to perform the appropriate calculations. Settlement calculation groups associated with settlement subscription types are those groups considered valid for settlement subscriptions of that type.

Settlement calculation groups can also be referenced by the Execute Usage Calculation Group calculation rule.

## Understanding Bulk Settlement Calculation Groups

Bulk settlement calculation groups are calculation groups intended for use with high volume calculations, such as UFE allocation.

Some specific differences between bulk settlement calculation groups and standard settlement calculation groups include:

- Bulk settlement calculation groups reference only a single settlement calculation rule.
- Execution of bulk settlement calculation groups is performed by batch execution only, typically within a set of calculations performed during settlement transaction processing.
- Bulk settlement calculation groups use the “Settlement Bulk Calculation Group” business object (D1–SettlementBulkCalcGroup).
- Bulk settlement calculation groups have a **Calculation Group Class** of “Settlement Bulk Calculation”.

## Configuring Settlement Calculation Groups

This portal is used to display and maintain a settlement calculation group.

Refer to Understanding Settlement Calculation Groups for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Calculation Group**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Calculation Group**: This zone displays basic information about a settlement calculation group.
- **Settlement Calculation Rules List**: This zone lists the settlement calculation rules belonging to the group. Not applicable to bulk calculation groups.


Copyright © 2026, Oracle and/or its affiliates.

- **Referencing Settlement Calculation Rules List**: This zone lists the settlement calculation rules that reference the group. Not applicable to bulk calculation groups.
- **Referencing Settlement Subscription Type List**: This zone lists the settlement subscription types that reference the group.
- **Referencing Settlement Subscription List**: This zone lists the settlement subscriptions that reference the group.

## Configuring Bulk Settlement Calculation Groups

The **Settlement Calculation Group** portal is used to display and maintain bulk settlement calculation groups.

Refer to Understanding Bulk Settlement Calculation Groups for more information about bulk settlement calculation groups and Configuring Settlement Calculation Groups for more information about creating and maintaining bulk settlement calculation groups.

The following apply when creating new bulk settlement calculation groups:

- Select “Settlement Bulk Calculation Group” from the **Settlement Calculation Group Business Object** drop-down list.
- Use a naming convention of some sort to identify bulk settlement calculation groups (the **Settlement Calculation Group** portal does not provide a way to search for groups based on **Calculation Group Class**). For instance, consider including “bulk” or “blk" in the **Description** of bulk settlement calculation groups.

# Settlement Calculation Rules

- Understanding Settlement Calculation Rules
- Configuring Settlement Calculation Rules

## Understanding Settlement Calculation Rules

Settlement calculation rules are standard and custom rules that perform settlement calculations, including calculating service point quantities, totaling consumption and usage for settlement units, application of losses, and allocation of unaccounted for energy (UFE). Settlement calculation rules are created for a specific settlement calculation group. For example, if you were configuring two settlement calculation groups and both included a specific usage calculation rule, you would need to create two instances of the settlement calculation rule, one for each group.

On almost every settlement calculation rule, the failure of the rule results in a settlement transaction exception and the Usage Transaction Exception Type for the failure can be configured on the rule. These usage transaction exception types can also be set to a specific Exception Severity:

- **Information**: Used to highlight minor issues, but not sufficient to cause the settlement transaction to be put into a failure state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues.
- **Issues**: Used to report a problem that will prevent the settlement transaction from being sent. Multiple "issue exceptions" can be created during settlement transaction processing. If at least one issue exists after all rules have been applied, the settlement transaction is transitioned to a failure state requiring review and approval.


Copyright © 2026, Oracle and/or its affiliates.

- **Terminate**: Used to report a severe issue that will cause the settlement calculation process to stop and the settlement transaction to be transitioned immediately to a failure state requiring review and approval. Only one terminate exception can be issued (as the first one causes calculation processing to stop for a settlement transaction). This should be used for cases where manual override / approval isn't accurate.

For a deeper functional understanding of settlement calculation, refer to the About Settlement Calculation section of the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide*.

## Configuring Settlement Calculation Rules

This portal is used to display and maintain a settlement calculation rule.

Refer to Understanding Settlement Calculation Rules, Understanding Settlement Calculation Groups, and Understanding Usage Transaction Exception Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Calculation Rule**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Calculation Rule**: This zone displays details for the selected settlement calculation rule, including parameters used when executing the rule
- **Eligibility Criteria List**: Lists the eligibility criteria defined for the rule, if applicable.

### Calculation Settlement Calculation Rules

The following is a list of the calculation settlement calculation rules provided as part of the base product. For more information on how each rule executes and can be configured, follow the link provided on the rule.

| Settlement Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Apply Math (Interval Data) | Interval | This rule is used to perform calculations on interval data and stores the results in the settlement transaction's service quantities. A variety of options are available on this rule that include defining the calculation type, variables to use, as well as the equation to use (math functions and expressions).<br>This rule provides aggregated usage for all selected interval measuring components (filter by TOU, SQI & UOM) associated to a settlement subscription.<br>This rule can also multiply total usage by a factor using a custom formula. |


Copyright © 2026, Oracle and/or its affiliates.

| Settlement Calculation Rule Name | Applicable Data Type(s) | Purpose |
|---|---|---|
| Array Math | Interval, arrays, and calculated quantities | This rule is used to perform settlement calculations on arrays and interval data and it stores the results in the settlement transaction's service quantities. |
| Market Award Allocation | Interval | This rule is used to perform settlement calculations to allocate load based on market awards, based on appropriate methods for dividing load out to wholesale contracts. |
| Measuring Component Set Calculation | Interval and calculated quantities | This rule is used to perform UFE calculations. |
| Vector and Service Quantity Math | Interval | This rule is designed to facilitate configuration of complex vector calculations. It is based on a series of underlying services with vectors configured as input to the calculations.<br>Typical uses:<br>Perform math using interval data. For example, take the difference between two curves, find max values, find coincident peaks, multiply a curve by a value, apply TOU maps, etc.<br>Define complex formulas using various interval curves, profile factor values or calculated service quantities (bill determinant values).<br>Support math functions: sin, cos, square root, etc.<br>Store derived curves in memory that can be used in subsequent calculations<br>Please note, this rule is not as efficient as other rules. |

### Decision-Making Settlement Calculation Rules

There are settlement calculation rules delivered as part of the base product that help with decision-making when executing the settlement calculation process. For more information on how each rule executes and can be configured, follow the link provided on the rule.


Copyright © 2026, Oracle and/or its affiliates.

| Settlement Calculation Rule Name | Purpose |
|---|---|
| Execute Usage Calculation Group | This rule performs a call to execute a separate settlement calculation group which includes execution of all settlement calculation rules within that group. |
| Exception Handler | This rule is used to terminate processing if exception count criteria specified in the rule is met. |

### Advanced Aside: Using Factors For Variables

A situation common in some implementations involves converting one unit of measure (UOM) to another. However, the conversion factor used in conversions of this type can differ based on many different types of criteria, such as the location of the service point or other characteristics. This type of calculation can be implemented as a settlement calculation rule that accumulates consumption for one UOM and converts the consumption to a different UOM by applying a factor to it.

Factors used for this purpose have a Factor Class of "Number," and use some unique rules:

- Number factors reference a characteristic type (with pre-defined values).


Copyright © 2026, Oracle and/or its affiliates.

- Number factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.

Factor values for a Number factor are effective-dated pairings of a characteristic value and a corresponding value. Because these pairings are effective-dated, the value returned from the factor can change over time for each characteristic value. At run time, the rule retrieves / derives the characteristic value for the factor's characteristic type and then finds the value associated with the respective characteristic value. Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute**: you could create a rule that retrieves a specific value based on the location of a service point.
- **Dynamic Attribute**: you could create a rule that retrieves a percentage value based on the amount the customer conserved as compared to the same period in the prior year, returning one value if the amount conserved is between 5% and 10%, another value if the amount conserved is between 10% and 20%, and a third value if the amount conserved is greater than 20%. The amount conserved is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and returns the appropriate value. In this example, if the amount conserved was anything less than 5%, no percentage value would be returned.
- Calculation Rules

## Calculation Rules

- Array Math
- Market Award Allocation
- Measuring Component Set Calculation

### Array Math

This rule is used to perform settlement calculations on arrays and interval data and it stores the results in the settlement transaction's service quantities. Arrays can be derived from either measuring component sets or calculated quantities. Interval data vectors can be derived from measuring component sets, individual measuring components, or calculated quantities. A variety of options are available on this rule that include defining variables to use for arrays and interval data vectors, the specific calculations to perform (leveraging math functions and expressions), and how to save the results.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-ARRAYMATH Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-ArrayMath


Copyright © 2026, Oracle and/or its affiliates.

### Market Award Allocation

This rule is used to perform settlement calculations to allocate load based on market awards, based on appropriate methods for dividing load out to wholesale contracts. This rule takes in a vector from the settlement transaction and divides the load to be allocated to various market awards for a configured market product set. The allocated amounts should be tied to the contract connected to the market award and then saved to the settlement transaction as a new array.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D2-AWARDALOC Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-MarketAwardAllocation

### Measuring Component Set Calculation

This rule is used to perform settlement UFE calculations.

This rule executes a batch program (D1-MCSCA, defined as a parameter on the Execute MC Set Calculation-Apply Rule algorithm), which in turn executes the Perform MC Set Calculation algorithm to perform the calculation. The rule configuration includes definition of variables and the equation used for UFE calculations. Variables can be based on specified measuring component sets, aggregation measuring components, or calculated quantities.

This rule should be used ONLY with Bulk Settlement Calculation Groups.

> **Note**
>
> Additional detail on the logic of this rule can be found in the Detailed Description of the D1-EXEMCSCAL Algorithm Type.
>
> For help with the meaning of specific configuration fields, refer to the embedded help on the screen when adding or editing the rule.

**Business Object**

D2-MCSetCalculation

# Settlement Item Types

- Understanding Settlement Item Types
- Configuring Settlement Item Types


Copyright © 2026, Oracle and/or its affiliates.

## Understanding Settlement Item Types

Settlement Item Types define specific types of end-customer settlement accounts.

Settlement item types define the following attributes:

- **Settlement Item Business Object**: The business object to use create settlement items of this type.
- **Settlement Billed Usage Business Object**: The business object to use to capture billed usage for a settlement account.
- **Valid Service Point Types**: Valid service point types for the settlement item type.

> **Note**
>
> Settlement item types are similar to usage subscription types used in Meter Data Management, and are based on the Usage Subscription Type maintenance object (D1-USTYPE).

## Configuring Settlement Item Types

Refer to Understanding Settlement Item Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Item Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Type List**: This zone lists all settlement item types. Broadcast a record to display the details of the selected record.
- **Settlement Item Type**: This zone displays details for the selected settlement item type.

# Settlement Item Quantity Types


- Understanding Settlement Item Quantity Types
- Configuring Settlement Item Quantity Types


## Understanding Settlement Item Quantity Types

Settlement item quantity types define types of quantities that can be stored for a settlement item. These quantity types are often infrequently calculated descriptors of a given customer and how they relate, at an aggregate level, to either the customer base as a whole or their particular rate class.

Settlement item quantity types use the following parameters:

- **Service Type**: The service type (electric, gas, water, etc.) for settlement item quantities of this type.
- **Settlement Item Quantity Business Object**: The business object used for settlement item quantities of this type.



Copyright © 2026, Oracle and/or its affiliates.

- **Settlement Item Quantity Identifiers**: Value identifiers related to the current settlement item quantity type (used to provide shorthand descriptions of the various types of values measured by settlement item quantities of this type).

## Configuring Settlement Item Quantity Types

You use the **Settlement Item Quantity Type** portal to display and maintain settlement item quantity types.

Refer to Understanding Settlement Item Quantity Types for more information.

You can access the portal by selecting **Admin**, then **Settlement**, then **Settlement Item Quantity Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Settlement Item Quantity Type List**: This zone lists all settlement item quantity type records. Broadcast a record to display the details of the selected record.
- **Settlement Item Quantity Type**: This zone displays details for the selected settlement item quantity type.

# Market Contract Types


- Understanding Market Contract Types
- Configuring Market Contract Types


## Understanding Market Contract Types

Market Contact Types define specific types of market contracts used in settlement processing.

Market contract types define the following:

- The market associated with market contracts of this type
- The pricing category for market contracts of this type
- Whether market contracts of this type have Unique Contract Buyers. When set to "Yes", market contracts of this type are validated to ensure that each buyer only has one valid contract at a point in time for the market contract type.

## Configuring Market Contract Types

Refer to Understanding Market Contract Types for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market Contract Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Contract Type List**: This zone lists all market contract type records. Broadcast a record to display the details of the selected record.
- **Market Contract Type**: This zone displays details for the selected market contract type.



Copyright © 2026, Oracle and/or its affiliates.

# Market Products


- Market Product Sets
- Market Product Types


## Market Product Sets


- Understanding Market Product Sets
- Configuring Market Product Sets


### Understanding Market Product Sets

Market Product Sets define the highest level grouping for a set of products that will be processed together. Market Product Sets are used by the Market Award Allocation settlement calculation rule as a way to gather a set of products and their related awards for division of load into an array of interval curves / vectors.

Market product sets define the following:

- The market associated with the product set
- The service type (electric, gas, etc.) of the product set

### Configuring Market Product Sets

Refer to Understanding Market Product Sets for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market Product Set**. You are brought to a query portal with options for searching. Once your record has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Product Set**: This zone displays details for the selected market product set.

## Market Product Types


- Understanding Market Product Types
- Configuring Market Product Types


### Understanding Market Product Types

Market Product Types define types of products used in settlement processing.

There are two primary types of market product types, each based on a specific business object:



Copyright © 2026, Oracle and/or its affiliates.

- **Market Product Type**: Used for market product types whose award allocation is based on the configuration of the Market Award Allocation settlement calculation rule.
- **Formula-Based Market Type**: Used for market product types with more complex award allocation. This business object provides the ability to configure complex conditional calculations that take in both interval and scalar variables.

Market product types are defined by the following:

- **Division**: The division associated with the market product type.
- **Number of Decimals**: The number of decimal values used for market award quantities based on market products of this type.
- **Product Business Object**: The market product business object used for market products of this type.
- **Award Business Object**: The market award business object used for market awards based on market products of this type.
- **Calculation Details**: Applicable to Formula-Based Product Types only. Details used in performing formula-based calculations for market products of this type. Details include:
  - **Calculation Pre-Processing**: One or more algorithms to be executed prior to performing calculations for products of this type. These can include validation algorithms used to ensure the data being used by the calculations is valid and algorithms that retrieve, calculate, and store values used later in the calculation.
  - **Calculation Inputs**: Input variables (including Fixed Variables, Award Adhoc Characteristics, Factor Numbers, and Calculated Quantities) and Input Vectors (Fixed Vector Variables, Calculated Quantities, and Factor Profiles) that will be used in the calculations performed for products of this type.
  - **Calculations**: One or more calculations (based on the Calculation Inputs) to be performed for products of this type. Calculations can be based on Simple or Conditional formulas.
  - **Calculation Post-Processing**: Details of how the results of the calculations are saved and used in later processing. Options include:
    * **Save Results to Calculated Quantity**: Defines how calculation results are saved as service quantities on the Settlement Transaction that initiated the market product calculation and, if applicable, assigned to market contracts.
    * **Save Results to Characteristics**: Defines how calculation results are saved as characteristic values on the Settlement Transaction that initiated the market product calculation.

## Configuring Market Product Types

Refer to Understanding Market Product Types for more information.

You can access the portal by selecting **Admin**, then **Settlement Market**, then **Market Product Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Market Product Type List**: This zone lists all market product type records. Broadcast a record to display the details of the selected record.
- **Market Product Type**: This zone displays details for the selected market product type.


Copyright © 2026, Oracle and/or its affiliates.

## Settlement Module Configuration

Oracle Utilities Market Settlements Management shares its application environment with other Oracle Utilities meter solution products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway. The default installation of any of these products provides access to all menu items and functionality provided with these products.

To configure the system such that only menu items and functionality used by Oracle Utilities Market Settlements Management is available, application users should be assigned to the following set of user groups:

- Meter Stack Common Services
- Meter Stack Common Batch Services*
- Settlement Services
- Settlement Batch Services*

*These user groups should be restricted to only those users who will be executing batch processing.

Refer to Defining Users in the *Oracle Utilities Application Framework Administrative User's Guide* for more information about setting up users.

> **Note**
>
> There is a Feature Configuration option that disables menu items and functionality used by Oracle Utilities Market Settlements Management even after assigning users to the above user groups. To check if this option is enabled, navigate to the **Feature Configuration** portal by clicking the **Menu** icon, then selecting **Admin Menu**, then **General**, and then **Feature Configuration**. In the **Feature Configuration Search** zone, select "Module Configuration" from the **Feature Type** dropdown list and click **Search**. Click the "Module Configuration" **Description** link and check to see if there is a "Turned Off" option defined in the **Options** zone with a value of "SE". If so, delete that option. Note you may have to flush the server cache (see **Caching Overview** in the *Framework Administrative User Guide*) after after deleting this option before the menu items and functionality are available.

# Additional Independent Modules

- Aggregation
- Consumption Synchronization
- Dashboards
- Data Extracts
- Information Lifecycle Management (ILM)
- Measurement Reprocessing
- Outage Storm Mode
- Running Initial Periodic Estimation


Copyright © 2026, Oracle and/or its affiliates.

# Aggregation


- Standard Aggregation
- Dynamic Aggregation


## Standard Aggregation


- Configuring an Out-of-the-box Aggregation
- Understanding an Example Out-of-the-box Aggregation
- Creating a New Custom Aggregation


### Configuring an Out-of-the-box Aggregation

Aggregation calculations should be run on an as needed basis. This can include running the following batch processes:

- Scanning for new aggregation dimension (D1-ADS): This process is applicable if the system is configured to use aggregation dimension scanners to detect new aggregation dimensions (such as a service point referencing a new transformer for which an aggregator measuring component doesn't currently exist)
- Performing aggregation calculations (D2-AGG): this batch handles the summarization of measurement values in order to create aggregated measurements.

Note that aggregation calculations should precede usage transaction processing if aggregated values serves as input to the calculation of bill determinants.

Refer to the About Aggregations section of the *Oracle Utilities Meter Solution Business User Guide* for more information on this functionality. Refer to Understanding Measuring Component Types for information on Measuring Component Types provided for aggregation.

### Understanding an Example Out-of-the-box Aggregation

The Oracle Utilities Meter Data Management base package includes an aggregation that aggregates measurement quantities for constituent measuring components based on postal code and service type dimensions. The table below outlines the types of objects used in this aggregation, based on the steps outlined above), and the specific objects for each type.

| Dependency Order | Object Type | Base Package Example |
|---|---|---|
| 1 | Aggregator Measuring Component Business Object | D2-Aggregator (Aggregator - Postal and Service Type) |
| 2 | Aggregator Measuring Component UI Maps | Display: D2-AggMCDisp (Service Type and Postal Aggregator-Display)<br>Maintenance: D2-AggMCMaint (Service Type and Postal Aggregator-Maintenance) |



Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| 3 | Aggregator Business Object Info Algorithm | D2-AMC-INFO (Service Type and Postal Aggregator - Information) |
| 4 | Find Constituent Measuring Components Algorithm | D2-DET-CMC (Find Constituent Measuring Components Based on Service Type and Postal) |
| 5 | Measuring Component Type | D2-AggregatorType (Aggregator Type) |
| 6 | Query Zone for Consumption Statistics Portal | D2-AGGMCQRY (Aggregator Search) |
| 7 | Dimension Scanner Activity Business Object | D2-ActivityAggDimScanner (Aggregator Creator - Postal / Service Type) |
| 8 | Dimension Scanner Activity UI Maps | Display: D2-AggDimScannerActDisp (Aggregator DS Activity-Display)<br>Maintenance: D2-AggDimScannerActMaint (Aggregator DS Activity-Maintenance) |
| 9 | Dimension Scanner Activity Business Object Info Algorithm | D2-ADS-INFO (Aggregator Dimension Scanner Information) |
| 10 | Enter Algorithm for Scan State | D2-CRE-AGGMC (Aggregator MC Creation for Post Code and Service Type) |
| 11 | Activity Type | D2-ActivityTypeAggDimScanner (Aggregator Dimension Scanner Activity Type) |

Refer to the About Aggregations section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

## Creating a New Custom Aggregation

This section describes the overall process for creating a new custom Aggregation.

Refer to the About Aggregations section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

**Execute the following steps:**

1. Create a business object for the aggregator measuring component. This will flatten the dimensional value(s) into searchable characteristics. Whether this business object is a parent or a child of another aggregator business object depends on when periodic aggregation should occur:
   a. If you want the periodic aggregation to occur when another aggregation occurs, it can be a child business object (meaning that it inherits the lifecycle (and therefore the deferred monitor) of the parent)
   b. If you want to schedule its periodic aggregation independently from other aggregation business objects, this must NOT be a child business object as it will require its own


Copyright © 2026, Oracle and/or its affiliates.

deferred monitor (and deferred monitors can only be defined on parent business objects)

2. Create UI maps for the aggregator business object as follows:
   a. One to display the aggregator measuring component (Display)
   b. One to allow user to change / add a new one (Maintenance)

> **Note**
>
> A newer alternative to creating UI Maps would be to use UI Hints directly within the Business Object.

3. Create an info plug-in for the aggregator business object that concatenates together its dimension types and values.
4. Create a "Find Constituent Measuring Components" algorithm and plug it on the aggregator business object. This will be passed the aggregator measuring component and the from and to date/times. It will insert the constituent measuring component IDs and the respective from / to date-time of each onto a temporary table.
5. Create a measuring component type instance and reference the new aggregator measuring component business object (as well as the types of constituent measuring component types that should be aggregated).
6. Create a query zone for Consumption Statistics search to allow users to find the aggregator measuring component.

**Optional steps:**

1. Create a business object for the dimension scanner activity. This should be a child business object of the base package dimension scanner business object.
2. Create UI maps for the activity business object, as follows:
   a. One to display the dimension scanner activity (Display).
   b. One to allow users to change/add a new one (Maintenance).

> **Note**
>
> A newer alternative to creating UI Maps would be to use UI Hints directly within the Business Object.

3. Create an info plug-in that will describe what it scans.
4. Create an Enter algorithm on the Scan state that finds distinct combinations of the dimensional values and creates new aggregator measuring components when new ones are detected.

> **Note**
>
> You can reuse the base package deferred monitor **Aggregation Dimension Scanner Monitor** (D1-ADS) batch control .

# Dynamic Aggregation


Copyright © 2026, Oracle and/or its affiliates.

- Dynamic Aggregation Configuration Overview
- Aggregation Master Configuration
- Data Sources
- Dynamic Aggregation Measuring Component Types
- Aggregation Groups
- Measuring Component Sets
- Configuring Dynamic Aggregation Step-by-Step
- Extending Dynamic Aggregation


## Dynamic Aggregation Configuration Overview

Standard aggregation processing uses specific algorithms configured to work with a specified set of dimensions. For example, the base package Service Type and Postal aggregation uses specific algorithms for dimension scanning and finding constituent measuring components.

In contrast, dynamic aggregation uses dynamic queries for the dimension scanning, find constituents, and aggregation processes. These dynamic queries are based on configuration of administrative data, including:

- **Data Sources** define the source of data to be aggregated, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or other sources of data.
- **Aggregation Measuring Component Types** define the most important properties of aggregation measuring components used to store aggregated data.
- **Aggregation Groups** define the ordering of a series of related aggregations and the schedule of aggregation.
- **Measuring Component Sets** define the dimensions and criteria by which aggregation will be performed.

## Aggregation Master Configuration

The Aggregation Master Configuration defines common parameters used with aggregation processing, including:

- Batch Controls used by the dimension scan and aggregation process, for both scheduled and adhoc execution.
- Details for how the system compares interval and billing data for the same set of customers in order to use only one source or other.

## Data Sources


- Understanding Data Sources
- Configuring Data Sources
- Configuring Data Source Template SQL


### Understanding Data Sources



Copyright © 2026, Oracle and/or its affiliates.

Data Sources define the source of data to be aggregated, such as measurement data from usage subscriptions linked to a service point, badged or unbadged items, or measuring component sets. Data sources also define configurations used when generating dynamic queries used by aggregation processing.

### Data Source Classes

Data sources are defined by classes, each of which is based on a specific business object, and determines the type of configuration used by aggregation processing. More specifically, the data types configured on the data source are used when generating dynamic queries used by aggregation processing.

The table below lists the data source classes and the business object and configuration options for each. Consult the embedded help for more information about the specific configuration options used by each data source class.

| Data Source Class | Usage | Business Object | Configuration Options |
|---|---|---|---|
| Measurements from US Service Points | Used when aggregating data from measuring components linked to service points via a usage subscription. | D1-DataSourceMeasurements | Measuring Component Types |
| Badged Items | Used when aggregating data from badged items. | D1-BadgedItemsDataSource | Item Types |
| Billed Service Quantities | Used when aggregating data based on billed services quantities. | D1-DataSourceBilledSerQuantity | Usage Subscription Types |
| Measuring Component Sets | Used when aggregating data from measuring component sets. Data sources of this class are used with composite aggregation. | D1-DataSourceMCSet | Usage Subscription Types |
| Unbadged Items | Used when aggregating data from unbadged items. | D1-DataSourceUnbadgedItems | Item Types |
| Measurements from US Direct Links | Used when aggregating data from measuring components directly linked to a usage subscription. | D1-DataSourceMeasurementDirect | Measuring Component Types |
| Service Point Quantities | Used when aggregating data based on service point quantities. | D1-DataSourceSPQuantities | Service Point Quantity Types |

### Data Source Template SQL

Data sources also provide template structured query language (SQL) used when generating dynamic queries used by aggregation processing. This template SQL is used as the basis for dynamic queries, and is extended by additional SQL generated from the data source configuration as well as configuration defined for measuring component sets, measuring


Copyright © 2026, Oracle and/or its affiliates.

components, and measuring component types that reference the data source. Template SQL is provided via the Data Source SQL (D1-DataSourceSQLLookup) extendable lookup. Values for this extendable lookup are used as the **Option Value** for the following business object options on the data source business object.

- **Dimension Scanner SQL**: The template SQL used when dimension scanning. This SQL is extended by the criteria, dimensions, and individually managed items configured on the measuring component sets being processed.
- **Find Constituents SQL**: The template SQL used when finding constituents. This SQL is extended by attributes defined on the aggregator measuring components being processed.
- **Aggregation SQL**: The template SQL used when performing aggregation. This SQL is extended by the value identifiers defined on the measuring component type of the aggregator measuring components being processed.

## Configuring Data Sources

This portal is used to display and maintain data sources.

Refer to Understanding Data Sources for more information.

You can access the portal by selecting **Admin**, then **Aggregation**, then **Data Source**.

The following zones may appear as part of the portal's **Main** tab page:

- **Data Source List**: This zone lists all data source records. Broadcast a record to display the details of the selected record.
- **Aggregation Data Source**: This zone provides information about the selected data source.

## Configuring Data Source Template SQL

Template SQL used when generating dynamic queries used in aggregation processing is defined in the Data Source SQL (D1-DataSourceSQLLookup) extendable lookup. The base package data source business objects reference appropriate base package values from this extendable lookup.

For example, the **Aggregation SQL** option on the Measurements from US Service Points data source business object (D1-DataSourceMeasurements) references the “D1_MSRMT_FROM_US_SP_MCTYPE_SQL” value from the Data Source SQL extendable lookup.

If an implementation requires template SQL that differs from the base package, custom extendable lookup values can be created and referenced on the data source business object.

When creating custom template SQL, start from an existing extendable lookup value. Select the value that most closely matches your requirements and use the **Duplicate** function to create a copy, and edit the SQL as needed.

Use the following procedure to add a custom extendable lookup value to a data source business object.

1. Add a business object option of the appropriate type to the data source business object. For instance, if creating custom SQL for aggregation, add an **Aggregation SQL** option.
2. Enter a **Sequence** higher than the base package option.
3. Specify the custom extendable lookup value in the **Option Value** field.


Copyright © 2026, Oracle and/or its affiliates.

# Dynamic Aggregation Measuring Component Types


- Understanding Dynamic Aggregation Measuring Component Types
- Configuring Dynamic Aggregation Measuring Component Types


## Understanding Dynamic Aggregation Measuring Component Types

Like other measuring component types, dynamic aggregation measuring component types define the most important properties of a measuring component. In this case, they define the properties of aggregator measuring components.

Measuring component types used by dynamic aggregation use many of the same attributes as other types of measuring components, but some attributes are used differently. Some attributes used specifically by dynamic aggregation include:

**Measuring Component Business Object**: This should be an aggregator measuring component business object, such as one of the following:

- Aggregator Interval (D1-AggregatorInterval)
- Aggregator Scalar (D1-AggregatorScalar)
- Aggregator Customer Quantities (D1-AggregatorCustomerQuantity)
- Aggregator Items Measuring Component (D1-AggregatorItems)

**Measurement Business Object**: Aggregated measurements can be based on either the Measurement (D1-Measurement) business object or the Aggregation Measurement (D1-AggregationMeasurement) business object (based on the Aggregation Measurement maintenance object).

**Data Source**: The source of the data that will be aggregated and the specific type of data that should be aggregated. Note that the **Service Type** of the data source, as determined by its associated objects (e.g. Measuring Component Type, Device Type, Service Point Type), must match the **Service Type** for this measuring component type.

**Anchor Time**: A time used to define the start and end time of the business day for purposes of aggregation processing for aggregation measuring components of this type. In gas settlement implementations, this is used to define the start and end time for "Gas Day" calculations.

**Value Identifiers**: These store the values of UOM, TOU, and SQI that identify the measurements to be aggregated. For each value identifier either the **Value Calculation Method** or **Value Derivation Algorithm** must be defined (not both). These define how measurement values are calculated or derived during aggregation processing.

Refer to Understanding Measuring Component Types for more information about measuring component types.

## Configuring Dynamic Aggregation Measuring Component Types

The **Measuring Component Type** portal is used to display and maintain dynamic aggregation measuring components.

Refer to Understanding Dynamic Aggregation Measuring Component Types for more information about dynamic aggregation measuring components.



Copyright © 2026, Oracle and/or its affiliates.

You can access the **Measuring Component Type** portal from **Admin > Device > Measuring Component**.

Configuring Measuring Component Types

# Aggregation Groups

- Understanding Aggregation Groups
- Configuring Aggregation Groups

## Understanding Aggregation Groups

Aggregation Groups define the ordering of a series of related aggregations based on a set of configured Measuring Component Sets. Aggregation groups define the following attributes for aggregations performed for its Measuring Component Sets:

- The **Time Zone** for aggregated data
- **Aggregation Scheduling** can be based on either a Defined Lag, or a Processing Timetable.
  - **Defined Lag** scheduling uses the following parameters:
    - **Aggregation Cut Off Time**: The end time for aggregation calculations performed for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Group. This is used to ensure a consistent end time for aggregation periods. This is especially useful when aggregating other aggregations.
    - **Aggregation Lag**: The number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the aggregation horizon that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.
    - **Aggregation Horizon**: The number of days in the aggregation period for Aggregation Measuring Components for Measuring Component Sets associated with the Aggregation Grou. This reflects the time period during which there's a potential change in measurement data for one or more of the measuring components associated with Aggregation Measuring Components. This together with the Aggregation Lag is used to determine the start and end dates of an aggregation period.
  - **Processing Timetable** scheduling uses a Processing Timetable Type to drive aggregation schedule.

Aggregation groups can be used to perform aggregations for Analytical or Settlement purposes (defined by the Aggregation Category).

- Aggregation Group Runs

## Aggregation Group Runs


Copyright © 2026, Oracle and/or its affiliates.

Aggregation group runs represent individual aggregation executions for an individual aggregation group. Aggregation group runs are used to capture the relevant details for an aggregation and track the status of the execution of the aggregation for each of the Measuring Component Sets for the aggregation group.

Aggregation group runs for a specific aggregation group are listed in the **Aggregation Group Run** zone on the **Aggregation Group** portal. This zone displays the following for each aggregation group run:

- Run Number
- Run Type (Scheduled vs. Ad Hoc)
- Status (Pending, In Progress, Completed, or Error)
- Aggregation Horizon Start
- Aggregation Horizon End
- Details

Details for a specific aggregation group run can be viewed by broadcasting the run in the **Aggregation Group Run** zone. The following is displayed for the selected run in the **Aggregation Group Run Details** zone:

- Sequence
- Measuring Component Set
- Initiation Method
- Batch Run

#### Aggregation Group Runs and Anchor Time

Aggregation processing supports calculations based on an "Anchor Time". The Anchor Time is specified on dynamic aggregation measuring component types and defines the start and end time of the business day for purposes of aggregation processing. When processing aggregation for dynamic aggregation measuring components whose type specifies an **Anchor Time**, the Start Time and End Time of the calculation period should match the **Anchor Time** on the dynamic aggregation measuring component type.

## Configuring Aggregation Groups

This portal is used to display and maintain aggregation groups.

Refer to Understanding Aggregation Groups for more information.

You can access the portal by selecting **Admin**, then **Aggregation**, **Aggregation Group**.

The following zones may appear as part of the portal's **Main** tab page:

- **Aggregation Group List**: This zone lists all aggregation group records. Broadcast a record to display the details of the selected record.
- **Aggregation Group**: This zone provides information about a selected aggregation group. The **Record Actions** section of this zone includes the following options:
  - Click **Delete Measurements** to clear the measurements of all measuring component sets of the aggregation group for a specified time range. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "Measurement Only" (MSRMTONLY ) Purge Option.


Copyright © 2026, Oracle and/or its affiliates.

- Click **Delete All MC Set Data** to clear all data for all measuring component sets of the aggregation group. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "All" (ALL ) Purge Option.
- Click **Delete All Aggregation Runs** to clears all aggregation group runs of the aggregation group. This runs the Clean Up Aggregation Group Runs (D1-AGGRC) batch process.

See **Batch Processing** below for more information about the batch processes used by these options.

- **Aggregation Measuring Component Sets**: This zone lists measuring component sets associated with the selected aggregation group. Click **Resequence** to change the order of the measuring component sets in the aggregation group. Click **Add** to add a new measuring component set to the aggregation group.
  - Click the icon in the **Delete Measurements** column to delete the measurements of the measuring component set for a specified time range. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "Measurement Only" (MSRMTONLY ) Purge Option for the measuring component.
  - Click the icon in the **Delete MC Set Data** column to delete all data of the measuring component set. This runs the Clean up Aggregation Group (D1-CLAGP) batch process using the "All" (ALL ) Purge Option for the measuring component.

  See **Batch Processing** below for more information about the batch processes used by these options.
- **Aggregation Group Run**: This zone lists individual aggregation runs performed for the selected aggregation group. Broadcast a record to display the details of the selected record. This zone displays up to 10 records. If there are more than 10 records, use the **Previous** and **Next** buttons to navigate between pages of records. Click **Add** to execute an ad hoc aggregation group run.
  - Click the icon in the **Delete** column to delete the aggregation group run. This runs the Clean Up Aggregation Group Runs (D1-AGGRC) batch process for the aggregation group run. Not applicable for aggregation group runs currently "In Progress". See **Batch Processing** below for more information about the batch processes used by this option.
  - Click the icon in **Cancel** column to cancel an "In Progress" aggregation group run. Applicable only for aggregation group runs currently "In Progress".
- **Aggregation Group Run Details**: This zone displays details for a selected aggregation group run.
  - Click **Clear Batch Runs** to clear any batch runs that have been run. This clears dimension scan batch run and aggregation batch run details.
  - Click the icon in the **Restart** column to restart the aggregation group run. Available only for aggregation group runs in "Error" (see the **Aggregation Batch Run** column)

The following zones may appear as part of the portal's **Settlement Batch Run Statistics** tab page:

- **Settlement Batch Run Statistics**: This zone displays details regarding batch processes run for aggregation groups.

### Batch Processing

Several of the options available for maintaining aggregation group runs are based on specific batch processes. This section provides additional details regarding these.

**Clean up Aggregation Group (D1-CLAGP)**


Copyright © 2026, Oracle and/or its affiliates.

The Clean up Aggregation Group (D1-CLAGP) batch process can be used to purge aggregation group data and can perform either of the following:

- Clean up the aggregation group's measuring component set(s)' aggregator measuring components and their aggregated measurements
- Clean up a specific measuring component set's aggregator measuring components and their aggregated measurements

Parameters used by this batch process include the following:

- **Restrict by Aggregation Group** (restrictToAggregationGroup): (Required) The code for the aggregation group for which data is purged. This limits the clean up to aggregators linked to this aggregation group's measuring component sets.
- **Restrict by MC Set** (restrictToMCSet): The code for the measuring component set for which data is purged. This limits the clean up to the specific aggregation group's measuring component set. If not provided, all the aggregation group's measuring component sets will be included in the purge process.
- **Purge Option** (purgeOption): (Required) Specifies the mode in which data is purged. Options include:
  - MSRMTONLY: - Purge measurement data only
  - ALL: Purge all measurement data, any settlement items related to the measuring component, any attribute data snapshots related to the settlement item, attribute value related to the Settlement Item, any related measuring component participants (aggregators)
- **Start Date/Time** (startDateTime): The start date and time for the period to be deleted when **Purge Option** is MSRMT_ONLY. This must be provided in the format YYYY-MM-DD-HH.MM.SS.
- **End Date/Time** (endDateTime): The end date and time for the period to be deleted when **Purge Option** is MSRMT_ONLY. This must be provided in the format YYYY-MM-DD-HH.MM.SS.

**Clean Up Aggregation Group Runs (D1-AGGRC)**

The Clean Up Aggregation Group Runs (D1-AGGRC) batch process can be used to purge either a single aggregation group run or all aggregation group runs of an aggregation group. Note this process does NOT purge the corresponding data such as measurements, settlement items, and so. Purging data can be performed using the Clean up Aggregation Group (D1-CLAGP) batch process.

Parameters used by this batch process include the following:

- **Restrict by Aggregation Group** (restrictToAggregationGroup): (Required) The code for the aggregation group for which aggregation group runs are pursed. This limits the clean up to aggregation group runs linked to this aggregation group
- **Restrict by Aggregation Group Run** (restrictToAggregationGroupRun): The code for a specific aggregation group run that will be cleaned up.

## Measuring Component Sets

- Understanding Measuring Component Sets
- Configuring Measuring Component Sets
- Configuring Aggregation Criteria Source Types


Copyright © 2026, Oracle and/or its affiliates.

## Understanding Measuring Component Sets

Measuring component sets are used to define the dimensions and criteria by which aggregation will be performed.

A measuring component set's **Measuring Component Type** defines the Measuring Component and Measurement Type to be used in capturing aggregation results, as well as the aggregation functions that will be used on the source transnational data and the Data Source that will identify where that transnational data will be retrieved from.

Measuring component sets are associated with an **Aggregation Group**, which controls when and how aggregation is processed. This includes the sequence in which the aggregation group's measuring component sets are processed, as well as the method by which aggregation processing is initiated for each (this information can be found in the **General Processing** section).

A measuring component set's **Measuring Component Set Class** defines the way in which the measuring component set is used in the aggregation process. Measuring component set classes are based on the measuring component set's business object, and can include:

- **Aggregation Foundation** measuring component sets are used to aggregate transactional data, such as measurements from service points linked via usage subscriptions or billed service quantities.
- **Aggregation Composite** measuring component sets are used to aggregate data from a set of other aggregations (which can be based on either foundation or other composite measuring component sets).
- **Adhoc** measuring component sets are used to aggregate data based on an adhoc, user-defined set of measuring components.

See **Foundation and Composite Aggregation** in Dynamic Aggregation Processing for more information.

### Dimensions, Criteria, and Individually Managed Items

The **Dimensions** and **Criteria** defined for the measuring component set are used by the dimension scanning process to identify individual aggregation measuring components that should be created. Measuring components are also created for each **Individually Managed Item** for which data matches the item's configuration.

- **Dimensions** identify the attributes by which data should be aggregated. Dimensions can include characteristics, identifiers, market participants, and values from specific tables and columns. For example, aggregating data by loss profile code, service point type and service provider would require defining dimensions for a Loss Profile Code characteristic, the Service Point Type column from the Service Point table, and the Service Provider column from the Device table.
- **Criteria** define the ways in which the dimensions are evaluated through a series of inclusions, exclusions, or both. For example, to aggregate data for the "AGG-E-COM" and "AGG-E-RES" service point types, but NOT for devices from the "ELEC" service provider, criteria could be defined to include both of service point type values and exclude the "ELEC" service provider.
- The criteria provided for **Individually Managed Items** is used to identify specific customers that should be aggregated in isolation from all other data that matches the **Dimensions** and **Criteria** defined for the measuring component set.


Copyright © 2026, Oracle and/or its affiliates.

More specifically, the dimension scanning template SQL (derived from the data source) is extended by the **Dimensions**, **Criteria**, and **Individually Managed Items** configured on the measuring component set being processed.

Consult the embedded help for more information about the specific configuration options used by **Dimensions**, **Criteria**, and **Individually Managed Items**. The Source Type and Source Entity values are defined in the Aggregation Criteria Source Type (D1-CriteriaSourceTypeLookup) extendable lookup. Refer to Configuring Aggregation Criteria Source Types for more information about this extendable lookup.

The manner in which the criteria provided will be evaluated is defined in formulas configured in the **Criteria Processing** section. Consult the embedded help for more information about defining criteria processing formulas.

#### Load Distribution

The **Load Distribution** parameters can be used to distribute the load of the transactional data derived from the associated Data Source when generating Measuring Component Set participants. Load distribution is achieved in the system by adding another dimensional attributes on the aggregator measuring component based on a defined number of digits from either the service point ID or usage subscription ID. **Load Distribution** parameters include:

- **Distribute Load**: A flag that indicates ("Yes" or "No") if if the measuring component set's transactional data should be further distributed via an additional dimensional attribute.
- **Number of Digits to Use**: The number of digits from either the service point ID or usage subscription ID to use as an additional dimensional attribute when distributing the load. For example, if **Number of Digits to Use** is 2, the last 2 digits of either the service point ID or usage subscription ID will be analyzed to determine if the transactional data should be a constituent of the aggregator measuring component.
- **Load Distribution Managed By**: Specifies which identifier, either "Service Point" or "Usage Subscription", whose last x digits will be placed as an additional dimensional attribute on the aggregator measuring component.
- **Load Distribution Formula**: Identifies if further criteria are needed when applying load distribution. For example, if "postal code" is a dimensional attribute, and we only want to distribute the load for a specific postal code where population is denser, a Criteria can be specified and this formula can be used to identify how that Criteria will be evaluated. This formula is applicable only when the **Distribute Load** parameter is set to "Yes".

## Configuring Measuring Component Sets

This portal is used to display and maintain measuring component sets.

Refer to Understanding Measuring Component Sets for more information.

You can access the portal by selecting **Admin**, then **Aggregation**, **Measuring Component Set**.

The following zones may appear as part of the portal's **Main** tab page:

- **Measuring Component Set**: This zone displays details of a selected measuring component set.
- **Dimensions**: This zone lists the dimensions used to identify the attributes by which the aggregation should be analyzed. Note: Data in this zone is edited by clicking the **Edit** button in the **Measuring Component Zone**.


Copyright © 2026, Oracle and/or its affiliates.

- **Criteria**: This zone lists the criteria that limits the data to be aggregated through a series of inclusions, exclusions, or both. Note: Data in this zone is edited by clicking the **Edit** button in the **Measuring Component Zone**.
- **Individually Managed Items**: This zone lists the criteria used to identify specific customers that should be aggregated in isolation from all other data that matches the dimensions and criteria defined for the measuring component set. Note: Data in this zone is edited by clicking the **Edit** button in the **Measuring Component Zone**.
- **Measuring Component Set Participants**: This zone displays the measuring components that are in the Measuring Component Set Participant table for the measuring component set. For each measuring component, this zone displays the most recent and last measurement date/times.

## Configuring Aggregation Criteria Source Types

The options in the **Source Type** and **Source Entity** drop-down lists used when configuring Dimensions, Criteria, and Individually Managed Items are defined in the Aggregation Criteria Source Type (D1-CriteriaSourceTypeLookup) extendable lookup. The measuring component set business objects reference this extendable lookup.

Each of the base package Source Type values (Characteristic, Identifier, Market Participant, Table, and Entity Group) are records in this extendable lookup. The Source Entity values for each of these are defined in the corresponding record.

For example, the “Table” Source Type defines the base package “Table” Source Entity values (Device, Device Event, Measuring Component, Service Point, Usage Subscription, and Usage), along with criteria path details for each (based on data source class).

If an implementation requires Source Type or Source Entity values that differ from the base package, custom extendable lookup values can be created.

When for creating custom values for this extendable lookup, start from an existing extendable lookup value. Select the value that most closely matches your requirements and use the **Duplicate** function to create a copy, and edit as needed.

# Configuring Dynamic Aggregation Step-by-Step

This section outlines the steps involved when configuring dynamic aggregation.

### Create Device and Service Point Administrative Data

Many of the objects used in dynamic aggregation reference other device- and service point-related administration data, such as measuring component types, service point types, usage subscription types, etc. Make sure this data is in place before configuring entities used with dynamic aggregation.

### Create Data Sources

Create data sources for each source of data from which data will be aggregated. Note that implementations can have multiple data sources of each class with different data type configuration (Measuring Component Types, Item Types, Usage Subscription Types, etc.). For example, an implementation with 15 measuring component types might want to aggregate data differently based on subsets of measuring component types. In this case, each subset of measuring component types could be defined on a different data source.


Copyright © 2026, Oracle and/or its affiliates.

If applicable, create custom Template SQL extendable lookup values and add them to Data Source business objects as appropriate.

**Create Aggregation Measuring Component Types**

Create the aggregation measuring component types that will be used to define the attributes of the measuring components that will be created during aggregation processing.

Aggregation measuring component types reference an appropriate Data Source. Continuing the example above, each subset of measuring component types (each defined on a separate data source) would require a separate aggregation measuring component type.

**Create Aggregation Groups**

Create the aggregation groups that will be used to control aggregation processing, including aggregation groups for both foundation and composite aggregation processing.

Note: When initially creating aggregation groups, define only the basic data for each (name, category, and optional time zone). Add measuring component sets to aggregation groups when creating the latter in the next step.

**Create Measuring Component Sets**

Create the measuring component sets that will define the “buckets” by which the aggregated data will be grouped.

Measuring component sets reference aggregation measuring component types and aggregation groups.

Create custom Source Type records (if applicable).

**Revise Aggregation Groups (if needed)**

After creating measuring component sets, adjust the sequence of each within its aggregation group as appropriate. For example, this would be necessary if measuring component sets are created and added to aggregation groups in an order that differs from their intended sequence.

## Extending Dynamic Aggregation

The dynamic aggregation functionality provides extensive flexibility through configuration of data sources, measuring component types, and measuring component sets to handle complex aggregation requirements. However, some implementations may require functionality and configuration options beyond that provided in the base package. For instance, an implementation may need to aggregate data of a source and type not covered by the base package data sources or use different dimensions and criteria or queries for aggregation processing.

In circumstances like these, the dynamic aggregation functionality can be extended through use of the Oracle Utilities Application Framework’s Configuration Tools (see Configuration Tools) and other options. This section outlines some of the possible approaches to extending this functionality.

**Data Source Business Objects**: If an implementation requires aggregation of a data type of source beyond those provided with the base package data source business objects, custom business objects can be created to support this requirement. Key considerations when creating custom data source business objects include:


Copyright © 2026, Oracle and/or its affiliates.

- Configuration options such as “Sourced Measuring Component Types” or “Sourced Usage Subscription Types”
- Values for the “Aggregation SQL”, “Dimension Scanner SQL”, and “Find Constituents SQL” business object options from the Data Source SQL extendable lookup.
- Algorithms for the “Generate SQL For Data Source” system event that leverage the configuration options defined for the data source

**Data Source SQL Extendable Lookup Values**: If an implementation requires template SQL that differs from the base package, custom values for the Data Source SQL extendable lookup can be created and referenced on the data source business object. When creating custom template SQL, start from an existing extendable lookup value. Select the value that most closely matches the requirements and use the **Duplicate** function to create a copy, and edit the SQL as needed.

**Aggregation Measuring Component Type Business Objects**: If an implementation requires capturing data for measuring component types not provided in the base package, the base package aggregation measuring component type business objects can be extended as appropriate. Key considerations when creating custom measuring component type business objects include:

- Algorithms for the “Generate SQL For MC Type” system event that leverages the Value Identifiers and other information defined for the measuring component type

**Aggregation Measuring Component Business Objects**: If an implementation requires capturing data for measuring components not provided in the base package, the base package aggregation measuring component business objects can be extended as appropriate. Key considerations when creating custom measuring component business objects include:

- Algorithms for the “Generate SQL For Aggregator MC” system event that leverages the Value Identifiers and other information defined for the measuring component type
- Algorithms for the “Find Constituents” system event that uses the “Generate SQL for Aggregator MC” algorithm, data source configuration, and other configuration to find constituent data for aggregation
- Algorithms for the business object’s “Active” state, including data aggregation and value derivation

**Measuring Component Set Business Objects**: If an implementation requires capturing data for measuring component sets not provided in the base package, the base package aggregation measuring component set business objects can be extended as appropriate. Key considerations when creating custom measuring component set business objects include:

- Algorithms for the “Generate SQL For MC Set” system event
- Algorithms for the business object’s lifecycle states, as appropriate

**Aggregation Criteria Source Type Extendable Lookup Values**: If an implementation requires Source Type or Source Entity values for measuring component sets that differ from the base package, custom extendable lookup values can be created. When creating custom values for this extendable lookup, start from an existing extendable lookup value. Select the value that most closely matches the requirements and use the **Duplicate** function to create a copy, and edit as needed.

# Consumption Synchronization

- Configuring Consumption Synchronization


Copyright © 2026, Oracle and/or its affiliates.

# Configuring Consumption Synchronization

> **Note**
>
> Refer to Introduction to Consumption Sync for additional functional information about how consumption synchronization works.

Keeping consumption synchronized between two measuring components that meter the same quantity but at different frequencies is a complicated task. As such the configuration for this process is diffuse and requires settings across several key areas of the system to be aligned. This section will help guide you through the process of configuring consumption synchronization.

The consumption synchronization process is really a collection of processes within Oracle Utilities Meter Data Management that all work together to ensure that quantities between related channels remain consistent:

- Estimation VEE Rules: several rules align consumption between related channels. These rules allow estimations to be refined based on higher quality measurements from a related channel. These rules are core to the consumption synchronization process. In simple scenarios where a few intervals are missing for an interval measuring component these rules are all that are necessary for synchronization with the related channel (note: it requires that the related channels data for that same time period has already been processed through to final measurements). For more information on these rules please visit the following sections:
  - Interval Adjustment from Scalar
  - Scalar Calculation from Interval
  - Scalar Proration
  - Sum Check
  - Final Measurement Validation

  > **Note**
  >
  > Refer to About IMD Estimations for more information about these rules

- Periodic Estimation: both the interval and scalar variations of periodic estimation play an important role in consumption synchronization by ensuring that both channels of data are without gaps. In simple scenarios where one channel is missing data and the other is not periodic estimation is all that is require to produce synchronized consumption. It is important to note that this process itself does not perform estimations but rather it is responsible for creating estimation initial measurements to trigger the estimation VEE rules.
- Consumption Synchronization Activities: these activities work to fix alignment issues that occur when data for the related channels are processed out of order or when there are complex outage scenarios. These activities are created when higher quality data is received for one channel and the related channel has measurements that are eligible to be recalculated and adjusted to align the total consumption for the period across the two channels. It is important to note that this process itself does not perform estimations but


Copyright © 2026, Oracle and/or its affiliates.

rather it is responsible for creating estimation initial measurements to trigger the estimation VEE rules.

### Device Configuration Type

As a default the system will not generate consumption synch activities for related measuring components. Turning on consumption synchronization activities is done through a few key fields on the device configuration type:

- **Keep Consumption Reference MC in Sync**: defines if the related measuring components should be kept in sync. It can be configured to provide a one way synchronization from primary to secondary or a two way synchronization between primary and secondary.
- **Minimum Condition to Sync Primary MC**: when the secondary measuring component can initiate synchronization of the primary this provides an ability to limit those situations to when the incoming data is of a minimum quality.
- **Sum Check VEE Exception Type**: provides further ability to limit initiation of synchronization. When configured the synchronization activities will only be created when the initial measurement being processed encounters a VEE exception of the type configured. More specifically, a sum check VEE exception which will indicate that the two channels are out of synch by a minimum tolerance amount. This can be used to avoid synchronization either when there is no difference between the channels or when there is only a small difference between the channels.
- **Enable Extended Reconciliation**: defines if synchronization calculations should extend to cover the total contiguous period of consumption synchronization eligible final measurements across both channels in the relationship. When set to “Yes” consumption synchronization algorithms on initial measurements will execute a measuring component relationship level gap analysis and create a consumption synchronization for an entire gap that will impact both measuring components in the relationship.

> **Note**
>
> Additional detail about these fields can be found in the embedded help for the device configuration type.

### Register Auto-Read Measuring Component Type

The configuration available for the register auto-read measuring component type has an indirect impact on the consumption synchronization process. This configuration is primarily intended to allow for a register to re-evaluate previously created estimations when new more accurate readings are received even when no related measuring components exist. Where this has impact to the consumption synchronization process is that when a new scalar reading is received after a period of estimation it will result in the time period for that new initial measurement being expanded into the past. This is because any estimates prior to this new higher quality initial measurement will be logically removed and the start reading for the initial measurement will in turn be the last non-estimated measurement prior to the initial measurement. This creates an initial measurement that spans the entire period for which estimations exist and as a result when a consumption synchronization activity is generated it will result in that same period of time being re-evaluated on the related measuring component.

This functionality is controlled by a few key fields on the register auto-read measuring component type:

- **Ignore Estimates as IMD Start Reading**: controls whether estimates directly previous to newly received incoming initial measurement should be logically removed when that newly received initial measurement data is non-estimated.


Copyright © 2026, Oracle and/or its affiliates.

- **Flag Future Estimates as Do Not Use**: controls whether estimates that come directly after newly received incoming initial measurements should be logically removed when that newly received initial measurement data is non-estimated
- **Actuals or Corrections Initial Re-Estimation:** works in tandem with the above two fields. When either scenario results in measurements being logically removed this field, when turned on, will reset the last contiguous measurement date/time for the measuring component making it eligible once more for periodic estimation. With the end result being that those estimated measurements that were removed would recreated as new estimates.

### Measuring Components

In order for consumption synchronization to work it must know that two measuring components are related in a way that indicates they are measuring the same consumption. This is achieved by configuring the "Consumption Reference Measuring Component" field on the measuring component.

Not only does this establish the relationship but it also establishes which measuring component is considered to be primary and which is considered to be secondary.

A measuring component is considered to be secondary when it holds the relationship to the Consumption Check Measuring Component (the primary measuring component).

For example, consider the following related measuring components.

- Scalar Measuring Component: ER-SM-007 / 1 / Electric kWh Daily
- Interval Measuring Component: ER-SM-007 / 2 / Electric kWh 60min

If the scalar measuring component is the primary measuring component, it does NOT specify a "Consumption Reference Measuring Component", and the interval measuring component specifies the scalar measuring component as the "Consumption Reference Measuring Component".

If the interval measuring component is the primary measuring component, it does NOT specify a "Consumption Reference Measuring Component", and the scalar measuring component specifies the interval measuring component as the "Consumption Reference Measuring Component".

It is important to note that pending initial measurements for the secondary measuring component are processed by the D1-IMD batch process after the initial measurements for the primary measuring component. This ensures that the primary measuring component's final measurements will be available to the secondary measuring component's initial measurements VEE process to provide better quality estimations and validations.

### Initial Load and Manual Initial Measurement Algorithms

Key to the consumption synchronization process are the algorithms that reside on initial load and manual initial measurements for both scalar and interval measuring components. These algorithms contain logic to identify final measurements on a related channel that are consumption synchronization eligible (typically estimated).

These algorithms allow definition of:

- The consumption synchronization activity created
- The condition range that defines consumption synchronization eligible final measurements

Please refer to the algorithm type descriptions for more information:

| Algorithm Type | Description | Where used |
|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| D1-UPD-DTMC | Update Latest Measurement Date/Time on MC with Consumption Sync | D1-InitialLoadIMDInterval<br>D1-ManualIMDInterval |
| D1-UDTSCMCRE | Update Latest Measurement Date/Time on Scalar MC with Consumption Sync | D1-InitialLoadIMDScalar<br>D1-ManualIMDScalar |

### Consumption Synchronization Activities

The consumption synchronization activities are initiated by initial measurements and are responsible for generating the appropriate estimation initial measurements to re-evaluate any consumption synchronization eligible final measurements for the related measuring component being processed.

These activities can be associated to one or more initiating initial measurements and are able to handle a broad combination of time periods which are not required to be contiguous.

If any generated estimation initial measurement does not finalize processing will be halted and details about the failed estimation initial measurement will be provided.

The following table identifies the catalogue of consumption synchronization activities:

| Activity Type | Description | Activity Business Object |
|---|---|---|
| Related MC Consumption Sync - Scalar | This activity is generated to re-evaluate consumption sync eligible final measurements on scalar measuring components. One estimation initial measurement will be created per measurement needing to be re-evaluated. | D1-RelMCREScalar |
| Related MC Consumption Sync - Interval | This activity is generated to re-evaluate consumption sync eligible final measurements on interval measuring components. One estimation initial measurement will be created per contiguous set of interval measurements needing to be re-evaluated. Note the generated initial measurements will include non-consumption sync eligible final measurements which will help to feed into the estimation process to provide more accurate results. | D1-RelMCREInterval |
| Gap Period Consumption Sync | This activity will re-evaluate both measuring components in the relationship for a wider time period than the initiating initial measurement. Specifically the time period will be the total contiguous period of consumption sync eligible final measurements across both channels in the relationship. | D1-GapPeriodConsumpnSync |


Copyright © 2026, Oracle and/or its affiliates.

Each consumption synchronization activity has an algorithm that performs the core logic of the consumption synchronization process.

These algorithms allow definition of:

- The condition range that defines consumption synchronization eligible final measurements

Please refer to the algorithm type descriptions for more information:

| Algorithm Type | Description | Where used |
|---|---|---|
| D1-RE-SCMC | Scalar MC Consumption Sync | D1-RelMCREScalar |
| D1-RE-INTVMC | Interval MC Consumption Sync | D1-RelMCREInterval |
| D1-GAPCSYNC | Gap Period Consumption Sync | D1-GapPeriodConsumpnSync |

#### Periodic Estimation

The algorithm that initiates periodic estimation which is plugged in to the Smart Meter device business object has a key configuration setting related to consumption synchronization:

- **MC Type to Process First**: determines whether interval or scalar measuring components should be estimated first. This should be set to process whichever measuring component type is considered to be primary first.

Please refer to the algorithm type description for more information:

| Algorithm Type | Description | Where used |
|---|---|---|
| D1-PERESTM | Periodic Estimation | D1-SmartMeter |

#### Related Batch Controls

There are a few batches involved with Consumption Synchronization:

- **Related MC Consumption Sync** (D1-RMCRE): this processes any pending consumption synchronization activities.
- **Related MC Consumption Sync - Retry Act** (D1-RMCRR): this retries any consumption synchronization activities that fail to the Issue Detected state.

# Dashboards

- Configuring the Meter Operational Dashboard
- Configuring the System Health Check Dashboard
- Configuring the Service Order Management Dashboards
- Configuring the Product Use Metrics Dashboard

## Configuring the Meter Operational Dashboard

This section describes the process for configuring the Meter Operational Dashboard.

Refer to the Using the Meter Operational Dashboard section of the *Oracle Utilities Meter Solution Business User Guide* for more information on this functionality.

This dashboard leverages a method of pre-staging data known as Statistics Snapshots. See Understanding Statistics for more information on Statistics Snapshots.


Copyright © 2026, Oracle and/or its affiliates.

A method for tracking Performance Targets (also known as Service Level Agreements) is available within the Meter Operational Dashboard as well. To understand how to set up Performance Targets, refer to Understanding Performance Targets. The Performance Targets you configure will then be leveraged on the **Batch Performance** and/or **Database** tabs of the Meter Operational Dashboard.

Configuring the **Meter Operational Dashboard** includes the following:

### Master Configuration

You can access the Master Configuration portal by selecting **Admin**, then **General**, then **Master Configuration**.

Select the MDM Operational Dashboard Configuration.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field.

### General Admin Data

Create the following administrative data.

**Service Type:**

- **Service Type**: MISC
- **Description**: Miscellaneous

**Unit of Measure:**

- **Unit of Measure**: CT
- **Description**: Count
- **Service Type**: Miscellaneous
- **Decimal Positions**: 0
- **Allowed on Measuring Component**: Yes
- **Measures Peak Quantity**: No
- **Magnitude**: 1
- **Base Unit of Measure**: BLANK

### Activity Types for Payload Statistics

Configure the following payload statistics activity types:

- Payload Statistics
- Payload Error Notification
- Payload Summary

See Configuring Activity Types for more information.

See Processing Statistics in the *Business User Guide* for more information about payload statistics processing.

### Measuring Component Types for Payload Statistics

Configure the following measuring component types:


Copyright © 2026, Oracle and/or its affiliates.

### Initial Measurements

| Field / Section | Value |
|---|---|
| Measuring Component Type | STATS-IMD |
| Description | IMD Upload Statistics Aggregator |
| Measuring Component Type Business Object | Activity Upload Statistic Aggregator Type |
| Measuring Component Business Object | Payload Statistics Aggregator - IMD |
| Measurement Business Object | Measurement |
| Service Type | Miscellaneous |
| Statistic Type | Initial Measurements |
| Display Parcel Size | 24:00:00 |
| Aggregation | Aggregation Horizon: 5<br>Aggregation Lag: 2 |
| Valid Activity Types to Aggregate | Activity Type: Payload Statistics |
| Value Identifiers | |

| Value Identifier Type | Short Hand Description | Unit of Measure |
|---|---|---|
| Measurement | Total Number of IMDs | Count |
| Value 01 | Count of Outstanding IMDs | Count |
| Value 02 | Count of IMDs in Error | Count |
| Value 03 | Count of Pending IMDs | Count |
| Value 04 | Count of Completed IMDs | Count |

| | |
|---|---|
| Display Configuration | Hours of Chart Data to Display: 24 |

### Device Events

| Field / Section | Value |
|---|---|
| Measuring Component Type | STATS-EVENT |
| Description | Event Upload Statistics Aggregator |
| Measuring Component Type Business Object | Activity Upload Statistic Aggregator Type |
| Measuring Component Business Object | Payload Statistics Aggregator - Event |
| Measurement Business Object | Measurement |
| Service Type | Miscellaneous |
| Statistic Type | Events |
| Display Parcel Size | 24:00:00 |
| Aggregation | Aggregation Horizon: 5<br>Aggregation Lag: 2 |
| Valid Activity Types to Aggregate | Activity Type: Payload Statistics |


Copyright © 2026, Oracle and/or its affiliates.

| Field / Section | Value |
|---|---|
| Value Identifiers | <table><tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr><tr><td>Measurement</td><td>Total Number of Events</td><td>Count</td></tr><tr><td>Value 01</td><td>Count of Outstanding Events</td><td>Count</td></tr><tr><td>Value 02</td><td>Count of Events in Error</td><td>Count</td></tr><tr><td>Value 03</td><td>Count of Pending Events</td><td>Count</td></tr><tr><td>Value 04</td><td>Count of Completed Events</td><td>Count</td></tr></table> |
| Display Configuration | Hours of Chart Data to Display: 24 |

### Statistic Controls and Statistics Snapshots

Statistic Controls are the source for recording the transactions which have gone into exception/ discarded state or have a pending to-do associated with them. These records are then fetched by MDM Operational Dashboard's respective zones for Statistic and Trend display.

Statistic Snapshots capture the transactions which have gone into exception/discarded state or have a pending to-do associated with them at a specific time.

Statistic Controls are configuration records that record the daily transactions of Incomplete IMD by Status, VEE Exceptions by Type, Usage Transaction by Status, and Usage Transaction by To-Do.

Statistic Snapshots are captured daily via the Statistics Control Monitor (F1-STATS) batch process.

Configure the following Statistics Control records:

| Statistics Control | Description | Category | Status | Snapshot Frequency Hours | Retention Mode | Snapshot Business Object |
|---|---|---|---|---|---|---|
| INCOMPLETE-IMD-STATUS | Incomplete IMDs by Status | Operational | Active | 1.00 | Retain All Snapshots | Same as Description |
| INCOMPLETE-UT-STATUS | Incomplete Usage Transaction by Status | Operational | Active | 1.00 | Retain All Snapshots | Same as Description |
| INCOMPLETE-UT-TD-EXCP | Incomplete Usage To Dos by Exception | Operational | Active | 1.00 | Retain All Snapshots | Same as Description |
| INCOMPLETE-VEE-TD-EXCP | Incomplete VEE To Dos by Exception | Operational | Active | 1.00 | Retain All Snapshots | Same as Description |


Copyright © 2026, Oracle and/or its affiliates.

### Payload Statistics Aggregator Measuring Components

Create a measuring component for each head end system for each measuring component type. For example, assume the following head end systems are being used:

- MV90: Usage
- Landis+Gyr: Usage and Device Events

In this example, you would create the following measuring components:

- MV90 (STATS-IMD Measuring Component Type)
- L+G (STATS-IMD Measuring Component Type)
- L+G (STATS-EVENT Measuring Component Type)

| Field | MV90 Usage | L+G Usage | L+G Device Events |
|---|---|---|---|
| Measuring Component Type | IMD Upload Statistics Aggregator | IMD Upload Statistics Aggregator | Event Upload Statistics Aggregator |
| Status | Active | Active | Active |
| How To Use | Additive | Additive | Additive |
| Head End System | MV90 | L+G | L+G |

### Batch Processing

The Meter Operational Dashboard is configured to be refreshed daily. To refresh the data used in the dashboard, the following batch processes should be processed in sequence after daily payload processing from the head end systems is complete.

- **D1-PSACC** (Payload Statistics Accumulation - Monitor): Used to initiate aggregation and accumulation of payload statistics in payload statistics activities. This executes the Perform Activity Aggregation (D1-PRFACTAGG) algorithm in the activity's "Active" state and transitions the activity to the "Accumulate Statistics" state which executes the Accumulate Statistics Algorithm (D1-ACCSTATS) algorithm.

  Use the following parameters when running this batch process:

  – Measuring Component Maintenance Object (maintenanceObject): D1-ACTIVITY

  – Restrict by Batch Code (isRestrictedByBatchCode): true

- **D1-MC** (MC MO Periodic Monitor Process): Used to initiate aggregation of statistics for statistics aggregator measuring components. This transitions measuring components from the "Active" state to the "Aggregate" state, which executes the Aggregate IMD Statistics (D1-AGIMDSTAT) or Aggregation Event Statistics (D1-AGEVTSTAT) algorithms that aggregate measurement and/or device event statistics.

  Use the following parameters when running this batch process:

  – Maintenance Object (maintenanceObject): D1-MEASRCOMP

- **F1-STATS** (Statistics Control Monitor): Used to generate Statistic Snapshots based on Statistic Controls.

  Use the following parameters when running this batch process:

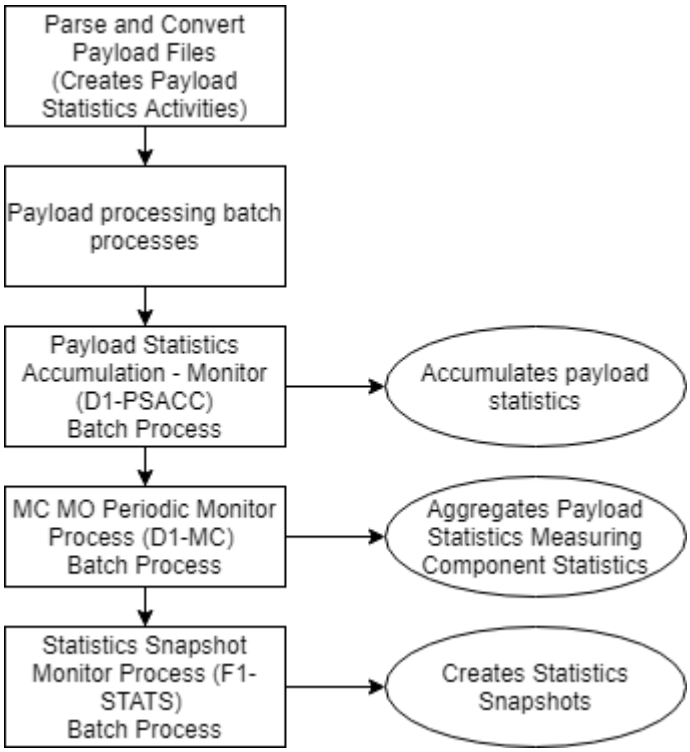
  – Maintenance Object (maintenanceObject): F1-STATS

  – Restrict by Business Object (restrictToBusinessObject): F1-Statistics

  – Restrict by Status Code (restrictToBOStatus): ACTIVE


Copyright © 2026, Oracle and/or its affiliates.

The flowchart below illustrates the order which payload processing and these batch processes should be executed.




### Populating the Dashboard Zones

The table below outlines the source of data for the various zones in the Meter Operational Dashboard.

| Dashboard Zone | Source of Data |
|---|---|
| IMD Loading Trend | Usage Payload Statistics Aggregator Measuring Components (which aggregate data from payload statistics activities)<br>(MV90 Usage and L+G Usage measuring components from the example above) |
| Device Event Loading Trend | Device Event Payload Statistics Aggregator Measuring Components (which aggregate data from payload statistics activities)<br>(L+D Device Events measuring component from the example above) |
| Incomplete IMDs by Status<br>Incomplete IMDs Trend | Incomplete IMDs by Status Statistics Control and Statistics Snapshots |
| VEE To Dos by Exception Type<br>VEE To Dos Trend | Incomplete VEE To Dos by Exception Statistics Control and Statistics Snapshots |
| Usage Transaction by Status<br>Usage Transactions Trend | Incomplete Usage Transactions by Status Statistics Control and Statistics Snapshots |
| Incomplete Usage To Dos by Exception<br>Usage Transactions To Dos Trend | Incomplete Usage To Dos by Exception Statistics Control and Statistics Snapshots |


Copyright © 2026, Oracle and/or its affiliates.

# Configuring the System Health Check Dashboard

This section describes the process for configuring the **System Health Check** tab of the **Meter Operational Dashboard**.

Refer to Using the System Health Check Tab in the *Business User Guide* for more information on this functionality.

This dashboard leverages a method of pre-staging data known as Statistics Snapshots. See Understanding Statistics for more information on statistics snapshots.

Configuring the**System Health Check** tab includes the following:

### Master Configuration

To access the **Master Configuration** portal select **Admin**, select **General**, and select **Master Configuration**.

Select the **System Health Check Configuration** master configuration.

You can also access this master configuration by clicking **Configure** on the **System Health Check** tab on the **Meter Operational Dashboard**.

Click the **Add** icon beside the record to configure for the first time. If a record has already been added, then click the **Edit** icon instead. Use the embedded help to guide you through the meaning of each configuration field.

This master configuration includes the following options:

- **IMD Trend Days**: specifies the horizon of collected data for IMD Processing and IMD Quality statistics snapshots. The number of days captured in each snapshot. Default 30 days.
- Chart Options:
  - **Processing**: Allows configuration of the colors used for IMD Records, Average Measurements, and Maximum Measurements data on the **IMD Processing Trend** graph.
  - **Quality**: Allows configuration of the colors used for VEE Exceptions, Average Measurements, and Maximum Measurements data on the **IMD Quality Trend** graph.

### Configure Statistic Control and Statistic Snapshots

Statistic controls are the source for capturing the numbers of initial measurement records, VEE exceptions, and average and maximum measurements displayed on the **Meter Health Check** tab.

Statistic snapshots capture the numbers for each metric. Statistic controls are configuration records that define the specific types of metrics to capture.

Statistic snapshots are captured periodically via the F1-STATS (Statistics Control Monitor) batch program.

Configure the following Statistics Control records:


Copyright © 2026, Oracle and/or its affiliates.

| Statistics Control | Description | Category | Status | Snapshot Frequency Hours | Retention Mode | Snapshot Business Object |
|---|---|---|---|---|---|---|
| Processing | IMD Processing Statistics - Hourly | Operational | Active | 24:00 | Retain All Snapshots | IMD Processing Snapshot - Hourly (D1-IMDProcessingHourly) |
| Quality | IMD Quality Statistics - Hourly | Operational | Active | 24:00 | Retain All Snapshots | IMD Quality Snapshot - Hourly (D1-IMDQuality Hourly) |
| IMD Control Seeder | IMD Control Seeder - Hourly | Operational | Active | 24:00 | Retain All Snapshots | IMD Control Seeder – Hourly (D1-IMDCtrlSeeder) |
| IMD Control | IMD Control - Hourly | Operational | Active | 24:00 | Retain All Snapshots | IMD Control – Hourly (D1-IMDCtrl) |

See Capturing Statistics for more information about statistics.

### Configure Statistics Control Business Object

The Statistics Control (F1-Statistics) business object should be configured to specify the "Statistics Control Monitor" batch control in the **Monitor Process** drop-down list on the "Active" state.

To configure this business object, do the following:

1. Open the Statistics Control (F1-Statistics) business object in the **Business Object** portal.
2. Navigate to the **Lifecycle** tab, and expand the "Active" state.
3. Select "Statistics Control Monitor" from the **Monitor Process** drop-down list.
4. Click **Save**.

### Batch Processing

The **System Health Check** tab is configured to be refreshed regularly (as frequently as daily). To refresh the data used in the dashboard, the F1-STATS batch process should be processed in sequence after other regularly schedule batch processing is complete to generate statistic snapshots based on statistic controls.

Use the following parameters when running this batch process:

- Maintenance Object (maintenanceObject): F1-STATS
- Restrict by Business Object (restrictToBusinessObject): F1-Statistics
- Restrict by Status Code (restrictToBOStatus): ACTIVE

# Configuring the Service Order Management Dashboards


Copyright © 2026, Oracle and/or its affiliates.

This section describes the process for configuring the **Service Order Operational Dashboard** and the **Service Order Trends Dashboard**.

Refer to the Using the Service Order Operational Dashboard and Using the Service Order Trends Dashboard sections of the *Business User Guide* for more information on this functionality.

Configuring the **Service Order Operational Dashboard** and **Service Order Trends Dashboard** includes the following:

### Configure Activity Type

Configure the following payload statistics activity type:

- Activity Statistics Aggregator Creator

Use the following procedure to create this activity type:

1. Select **Admin**, then **Communication**, then **Activity Type**. The **Activity Type** portal and **Activity Type List** zone open.
2. Click the **Add/Edit** icon for the "Activity Statistics Aggregator Creator" activity type in the **Activity Type List** zone.
3. Specify the following:
   - **Create Aggregators for Field Activity Statistics**: Yes
   - **Create Aggregators for Command Activity Statistics**: Yes
   - **Activity-Requesting Service Providers**: The external system(s) that sends service requests
   - **Exception Handling**: **To Do Type**, **To Do Role**, and other values as appropriate for your implementation.
4. Click **Save**.

See **Configuring Activity Types** for more information.

### Configure Measuring Component Types

Configure the following measuring component types for each service type in your implementation:

- Activity Processing Accumulation Aggregator
- Activity Aging Snapshot Aggregator
- Activity Processing Snapshot Aggregator
- Activity Processing Derived Snapshot Aggregator

The tables below provide details for these measuring component types based on the "Electric" service type from the demonstration database.

**Sub-Aggregator Measuring Component Types:**

| Field | Activity Processing Snapshot Aggregator | Activity Processing Derived Snapshot Aggregator |
|---|---|---|
| Measuring Component Type | ACT_PROC_SNAP_AGG | ACT_PROC_DER_SNAP_AGG |
| Description | Activity Processing Snapshot Aggregator | Activity Processing Derived Snapshot Aggregator |


Copyright © 2026, Oracle and/or its affiliates.

| Field | Activity Processing Snapshot Aggregator | Activity Processing Derived Snapshot Aggregator |
|---|---|---|
| Measuring Component Type Business Object | Activity Statistics Sub-Aggregator Type | Activity Statistics Sub-Aggregator Type |
| Measuring Component Business Object | Activity Processing Snapshot Aggregator | Activity Derived Snapshot Aggregator |
| Measurement Business Object | Measurement | Measurement |
| Service Type | Electric | Electric |
| Interval Size | 24:00:00 | 24:00:00 |
| Fallback Time Zone | US Eastern Time | US Eastern Time |
| Consumptive/Subtractive | Consumptive | Consumptive |
| Activity Aggregator Type | Activity Processing Snapshot | Activity Processing Derived Snapshot |
| Aggregation | Aggregation Horizon: 7<br>Aggregation Lag: 2<br>Aggregation Cut-Off Time: 12:00:00 | Aggregation Horizon: 7<br>Aggregation Lag: 1<br>Aggregation Cut-Off Time: 12:00:00 |


Copyright © 2026, Oracle and/or its affiliates.

<table>
<tr><th>Field</th><th>Activity Processing Snapshot Aggregator</th><th>Activity Processing Derived Snapshot Aggregator</th></tr>
<tr><td>Value Identifiers</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Created</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed</td><td>Count</td></tr>
<tr><td>Value 02</td><td>Error</td><td>Count</td></tr>
<tr><td>Value 03</td><td>Discarded</td><td>Count</td></tr>
</table>
</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Total</td><td>Count</td></tr>
<tr><td>Value 10</td><td>Discarded</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed</td><td>Count</td></tr>
<tr><td>Value 02</td><td>Error Total</td><td>Count</td></tr>
<tr><td>Value 03</td><td>Error in Field</td><td>Count</td></tr>
<tr><td>Value 04</td><td>Error in Command</td><td>Count</td></tr>
<tr><td>Value 05</td><td>Error in Orchestrator</td><td>Count</td></tr>
<tr><td>Value 06</td><td>In Progress Total</td><td>Count</td></tr>
<tr><td>Value 07</td><td>In Progress in Field</td><td>Count</td></tr>
<tr><td>Value 08</td><td>In Progress in Command</td><td>Count</td></tr>
<tr><td>Value 09</td><td>In Progress in Orchestrator</td><td>Count</td></tr>
</table>
</td></tr>
<tr><td>Sub-Aggregator Types</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Display Configuration</td><td>Hours of Chart Data to Display: 480<br>Default Measurement Condition: Regular</td><td>Hours of Chart Data to Display: 480<br>Default Measurement Condition: Regular</td></tr>
</table>

### Master Aggregator Measuring Component Types:

| Field | Activity Processing Accumulation Aggregator | Activity Aging Snapshot Aggregator |
|---|---|---|
| Measuring Component Type | ACT_PROC_ACC_AGG | ACT_AGN_SNAP_AGG |
| Description | Activity Processing Accumulation Aggregator | Activity Aging Snapshot Aggregator |


Copyright © 2026, Oracle and/or its affiliates.

| Field | Activity Processing Accumulation Aggregator | Activity Aging Snapshot Aggregator |
|---|---|---|
| Measuring Component Type Business Object | Activity Statistics Master Aggregator Type | Activity Statistics Master Aggregator Type |
| Measuring Component Business Object | Activity Processing Accumulation Aggregator | Activity Aging Snapshot Aggregator |
| Measurement Business Object | Measurement | Measurement |
| Service Type | Electric | Electric |
| Interval Size | 24:00:00 | 24:00:00 |
| Fallback Time Zone | US Eastern Time | US Eastern Time |
| Consumptive/Subtractive | Consumptive | Consumptive |
| Activity Aggregator Type | Activity Processing | Activity Aging Snapshot |
| Aggregation | Aggregation Horizon: 5<br>Aggregation Lag: 2<br>Aggregation Cut-Off Time: 12:00:00 | Aggregation Horizon: 5<br>Aggregation Lag: 2<br>Aggregation Cut-Off Time: 12:00:00 |


Copyright © 2026, Oracle and/or its affiliates.

<table>
<tr><th>Field</th><th>Activity Processing Accumulation Aggregator</th><th>Activity Aging Snapshot Aggregator</th></tr>
<tr><td>Value Identifiers</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Activity Count</td><td>Count</td></tr>
<tr><td>Value 10</td><td>Discarded Count</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed Count</td><td>Count</td></tr>
<tr><td>Value 02</td><td>Error Count</td><td>Count</td></tr>
<tr><td>Value 03</td><td>Error in Field</td><td>Count</td></tr>
<tr><td>Value 04</td><td>Error in Command</td><td>Count</td></tr>
<tr><td>Value 05</td><td>Error in Orchestrator</td><td>Count</td></tr>
<tr><td>Value 06</td><td>In Progress Count</td><td>Count</td></tr>
<tr><td>Value 07</td><td>In Progress in Field</td><td>Count</td></tr>
<tr><td>Value 08</td><td>In Progress in Command</td><td>Count</td></tr>
<tr><td>Value 09</td><td>In Progress in Orchestrator</td><td>Count</td></tr>
</table>
</td><td>
<table>
<tr><th>Value Identifier Type</th><th>Short Hand Description</th><th>Unit of Measure</th></tr>
<tr><td>Measurement</td><td>Average Completion Time</td><td>Count</td></tr>
<tr><td>Value 01</td><td>Completed Within Tolerance</td><td>Count</td></tr>
<tr><td>Value 02</td><td>12-19 Hours Late</td><td>Count</td></tr>
<tr><td>Value 03</td><td>20-27 Hours Late</td><td>Count</td></tr>
<tr><td>Value 04</td><td>28+ Hours Late</td><td>Count</td></tr>
</table>
</td></tr>
<tr><td>Sub-Aggregator Types</td><td>
<table>
<tr><th>Activity Aggregator Type</th><th>Measuring Component Type</th></tr>
<tr><td>Activity Processing Derived Snapshot</td><td>Activity Processing Derived Snapshot Aggregator</td></tr>
<tr><td>Activity Processing Snapshot</td><td>Activity Processing Snapshot Aggregator</td></tr>
</table>
</td><td>N/A</td></tr>
</table>


Copyright © 2026, Oracle and/or its affiliates.

| Field | Activity Processing Accumulation Aggregator | Activity Aging Snapshot Aggregator |
|---|---|---|
| Display Configuration | Hours of Chart Data to Display: 720 | Hours of Chart Data to Display: 480 |
| | Default Measurement Condition: Regular | Default Measurement Condition: Regular |

### Create Activity Aggregator Measuring Components

Create an activity aggregator measuring component for each activity type you wish to capture for each measuring component type you created above. For example, if you wanted to capture statistics for request orchestration activity types (such as Enable Service, Disable Service, and so on) for the above measuring component types, you would create a total of 24 measuring components, as follows:

| Measuring Component Type | Aggregation Dimensions — Activity Type | Aggregation Dimensions — Service Type |
|---|---|---|
| Activity Processing Accumulation Aggregator | Back to Back Orchestrator | Electric |
| Activity Aging Snapshot Aggregator | Back to Back Orchestrator | Electric |
| Activity Processing Snapshot Aggregator | Back to Back Orchestrator | Electric |
| Activity Processing Derived Snapshot Aggregator | Back to Back Orchestrator | Electric |
| Activity Processing Accumulation Aggregator | Cut for Non Payment Orchestrator | Electric |
| Activity Aging Snapshot Aggregator | Cut for Non Payment Orchestrator | Electric |
| Activity Processing Snapshot Aggregator | Cut for Non Payment Orchestrator | Electric |
| Activity Processing Derived Snapshot Aggregator | Cut for Non Payment Orchestrator | Electric |
| Activity Processing Accumulation Aggregator | Disable Service | Electric |
| Activity Aging Snapshot Aggregator | Disable Service | Electric |
| Activity Processing Snapshot Aggregator | Disable Service | Electric |
| Activity Processing Derived Snapshot Aggregator | Disable Service | Electric |
| Activity Processing Accumulation Aggregator | Enable Service | Electric |
| Activity Aging Snapshot Aggregator | Enable Service | Electric |
| Activity Processing Snapshot Aggregator | Enable Service | Electric |
| Activity Processing Derived Snapshot Aggregator | Enable Service | Electric |


Copyright © 2026, Oracle and/or its affiliates.

| Measuring Component Type | Aggregation Dimensions — Activity Type | Aggregation Dimensions — Service Type |
|---|---|---|
| Activity Processing Accumulation Aggregator | Exchange Meter | Electric |
| Activity Aging Snapshot Aggregator | Exchange Meter | Electric |
| Activity Processing Snapshot Aggregator | Exchange Meter | Electric |
| Activity Processing Derived Snapshot Aggregator | Exchange Meter | Electric |
| Activity Processing Accumulation Aggregator | Reconnect Service for Payment Orchestrator | Electric |
| Activity Aging Snapshot Aggregator | Reconnect Service for Payment Orchestrator | Electric |
| Activity Processing Snapshot Aggregator | Reconnect Service for Payment Orchestrator | Electric |
| Activity Processing Derived Snapshot Aggregator | Reconnect Service for Payment Orchestrator | Electric |

If you wanted to also capture statistics for smart meter commands or field task types, you would need to create activity aggregator measuring components for each individual command or field task type for each measuring component type.

If you are creating service order activities from multiple initiating systems (such as a customer information system and a network management system), you would need to create separate sets of activity aggregator measuring components for each initiating system.

### Configure the Service Order Management Master Configuration

You can access the Master Configuration portal by selecting **Admin**, then **General**, then **Master Configuration**.

Select the Service Order Management Master Configuration.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field.

### Create an Activity for each Measuring Component Type

Create an activity for each measuring component type created above. Based on the example above (from the demonstration database) you would create an activity for each of the following measuring component types for each service type:

- Activity Processing Accumulation Aggregator
- Activity Aging Snapshot Aggregator
- Activity Processing Snapshot Aggregator
- Activity Processing Derived Snapshot Aggregator

For example, if you configured measuring component types for both "Electric" and "Water" service types, you would create eight activities (two of each of the types listed above).

Use the following procedure to create these activities:


Copyright © 2026, Oracle and/or its affiliates.

1. Select **Menu**, then **Communication**, then **Activity**, then **Add**. The **Select Type** page opens.
2. Select "Activity Statistics Aggregator Creator" from the **Activity Type** drop-down list and click **OK**. The **Activity Statistics Aggregator Creator** page opens.
3. Select the appropriate measuring component type from the **Aggregator Measuring Component Type** drop-down list.
4. Select the appropriate service type ("E, "Water", and so on) from the **Service Type** drop-down list.
5. Click **Save**.

See Maintaining Activities in the *Business User Guide* for more information about creating and maintaining activities.

### Aggregation and Batch Processing

The Service Order Management dashboards are based on aggregated data, and are configured to be refreshed daily. To refresh the data used in the dashboard, the following aggregation and batch processes should be processed in sequence each day.

#### Dimension Scanning

- **D1-ADS** (Aggregation Dimension Scanner Monitor): Used to transition "Activity Statistics Aggregator Creator" activities (created above) to the "Scan" state which checks for aggregator measuring components based on each measuring component type for each combination of dimensions. If needed, this process creates additional aggregator measuring components.

Use the following parameters when running this batch process:

- Measuring Component Maintenance Object (maintenanceObject): D1-ACTIVITY
- Restrict by Batch Code (isRestrictedByBatchCode): true
- Restrict to Business Object (restrictToBusinessObject): D1-ActivityStatsAggtorCreator
- Restrict by Status Code (restrictToBOStatus): ACTIVE

Note: Dimension scanning can also be performed manually by clicking **Scan** on the individual activities in the **Activity** portal.

#### Aggregation

- **D1-ACTAG** (Activity Statistics Aggregation Monitor): Used to aggregate activity data (via aggregator measuring components) to be displayed in the Service Order Management dashboards.

Use the following parameters when running this batch process:

- Restrict by Batch Code (isRestrictedByBatchCode): true
- Restrict to Business Object (restrictToBusinessObject): Aggregator Measuring Component business object, based on the measuring component types configured earlier:
  - D1-ActivityAgingSnapAggtor (Activity Aging Snapshot Aggregator)
  - D1-ActivityProcsAggregator (Activity Processing Accumulation Aggregator)
  - D1-ActivityProcDerivedSnapAgg (Activity Derived Snapshot Aggregator)
  - D1-ActivityProcSnapAggtor (Activity Processing Snapshot Aggregator)
- Restrict by Status Code (restrictToBOStatus): ACTIVE


Copyright © 2026, Oracle and/or its affiliates.

- Override Maximum Errors (maxErrors): 1000

Note: Aggregation can also be performed manually by clicking **Update Statistics** on the **Activity Statistics** dashboard zone.

# Configuring the Product Use Metrics Dashboard

This section describes the process for configuring the **Product Use Metrics** dashboard used with Oracle Utilities applications, including Oracle Utilities Billing Cloud Service (BCS), Oracle Utilities Customer to Meter (C2M), Oracle Utilities Customer Care and Billing (CCB), Oracle Utilities Customer Care and Billing Cloud Service (CCSBCS), Oracle Utilities Customer Cloud Service (CCS), Oracle Utilities Meter Data Management (MDM), Oracle Utilities Meter Solution Cloud Service (MSCS), Oracle Utilities Work and Asset Management (WAM), and Oracle Utilities Work and Asset Cloud Service (WACS).

The Product Use Metrics dashboard displays customer counts such as device counts, channels of an active meters, number of billable services, head end device counts, and other similar data. The data presented on each dashboard is tailored to provide summary information for management. Refer to Product / Metrics for more information about this dashboard.

The dashboard contains a number of zones based on the product you're using. The table below lists the zones and the products for which they are available.

| Zone | Products | Uses Statistics Snapshots |
|---|---|---|
| Asset Count | BCS, C2M, CCS, WAM, WACS | No |
| Billable Services by CIS Division | BCS, C2M, CCB, CCBCS, CCS | Yes |
| Billable Services by SA Type | BCS, C2M, CCB, CCBCS, CCS | Yes |
| Billable Services by Service Type | BCS, C2M, CCB, CCBCS, CCS | Yes |
| Channel Counts | BCS, C2M, CCS, MDM, MSCS | Yes |
| Device Counts | BCS, C2M, CCS, MDM, MSCS | Yes |
| Head End Device Count | BCS, C2M, CCS, MDM, MSCS | Yes |
| Usage Subscription Customer Count | BCS, C2M, CCS, MDM, MSCS | Yes |
| User Count | BCS, C2M, CCS, WAM, WACS | No |

Several of the zones on this dashboard leverage a method of pre-staging data known as Statistics Snapshots. See Understanding Statistics for more information on Statistics Snapshots.

Configuring the **Product Use Metrics** dashboard includes the following:

### Configure Statistics Category Lookup Value

Statistic Controls (see below) reference a "Statistics Category" defined as a value for the Statistics Category lookup (STAT_CATEGORY_FLG). Create a value for this lookup as follows:

- **Field**: PUMT
- **Description**: Product Use Metrics
- **Java Field Name**: productUseMetrics
- **Status**: Active


Copyright © 2026, Oracle and/or its affiliates.

### Configure Statistic Control and Statistic Snapshots

Statistic Controls are the source for capturing the numbers of active usage subscriptions and, device counts per head end system.

Statistic Snapshots capture the numbers for each metric. Statistic Controls are configuration records that define the specific types of metrics to capture.

Statistic Snapshots are captured periodically via the F1-STATS (Statistics Control Monitor) batch program.

Configure the following Statistics Control records:

| Statistics Control | Description | Category | Status | Snapshot Frequency Hours | Retention Mode | Snapshot Business Object |
|---|---|---|---|---|---|---|
| HEAD-END-DEVICE-COUNT | Head End Device Count | Product Use Metrics | Active | 1.00 | Only Retain Most Recent Snapshot | Head End Device Count Snapshot |
| US-CUSTOMER-COUNT | Usage Subscription Customer Count | Product Use Metrics | Active | 1.00 | Only Retain Most Recent Snapshot | Usage Subscription Customer Count Snapshot |

See Capturing Statistics for more information about statistics.

### Configure Device Count and Channel Count Product Metric Types

Product Metric Types define configuration related to a given metric, including which cloud offerings it applies to and the product metric snapshot business object to use when capturing the metric. Corfirm the following Product Metric Types are defined.

| Product Metric Type | Description | Snapshot Business Object | Extrenal Name(s) |
|---|---|---|---|
| D1-CHANNELCOUNTMETRICTYPE | Channel Count Metric Type | Channel Count Metric (D1-ChannelCountMetric) | CHANNEL COUNT TOTAL<br>DAILY READ COUNT TOTAL |
| D1-DEVICECOUNTMETRICTYPE | Device Count Metric Type | Device Count Metric (D1-DeviceCountMetric) | DEVICE COUNT |

See Capturing Statistics for more information.

### Configure Device Count and Channel Count Snapshot Business Objects and Algorithms (Optional)

By default, the Device Count Metric and Channel Count Metric business objects check for active usage subscriptions when deriving the total count of active devices and measuring components. However, in some implementations this is not required or appropriate.

The usage subscription check can be disabled if needed by setting the "Exclude US Check (Y or N)" parameter on the "Create XXXXX Count Metric Shapshot" Enter algorithms on the


Copyright © 2026, Oracle and/or its affiliates.

Complete state of the Snapshot business objects to "Y" (this parameter is set to "N" by default).

The table below lists the snapshot business objects and their respective algorithms.

| Snapshot Business Object | Count Snapshot Algorithm |
|---|---|
| Channel Count Metric (D1-ChannelCountMetric) | Create Channel Count Metric Snapshot (D1CHLCNTSNP) |
| Device Count Metric (D1-DeviceCountMetric) | Create Device Count Metric Snapshot (D1DEVCNTSNP) |

### Configure Customer Metrics Dashboard Configuration Master Configuration (Optional)

The Customer Metrics Dashboard Configuration master configuration defines properties used with the Billable Services zones in the Products Use Metrics dashboard, including:

- Water Service Types and Waste Water Service Type
- Excluded Billable SA Types
- Excluded Debt Classes
- Service Type Colors
- CIS Division Colors

Use the **Master Configuration** portal to configure this master configuration as appropriate for your implementation.

### Configure Statistics Control Business Object

The Statistics Control (F1-Statistics) business object should be configured to specify the "Statistics Control Monitor" batch control in the **Monitor Process** drop-down list on the "Active" state.

To configure this business object, do the following:

1. Open the Statistics Control (F1-Statistics) business object in the **Business Object** portal.
2. Navigate to the **Lifecycle** tab, and expand the "Active" state.
3. Select "Statistics Control Monitor" from the **Monitor Process** drop-down list.
4. Click **Save**.

### Batch Processing

The **Product Use Metrics** dashboard is configured to be refreshed periodically. To refresh the data used in the dashboard, the following batch processes should be processed in sequence after other regularly schedule batch processing is complete.

- **F1-STATS** (Statistics Control Monitor): Used to generate Statistic Snapshots based on Statistic Controls.

Use the following parameters when running this batch process:

- Maintenance Object (maintenanceObject): F1-STATS
- Restrict by Business Object (restrictToBusinessObject): F1-Statistics
- Restrict by Status Code (restrictToBOStatus): ACTIVE


Copyright © 2026, Oracle and/or its affiliates.

# Data Extracts

This section describes functionality that support extract and export of data for use in external applications and systems.

- Configurable Consumption Extracts
- Maintenance Object Data Export
- Service Point Consumption Extracts
- Consumption Extract Type
- Analytics Configuration
- Migrating Data Extract Processing from Legacy Measurement Processing to Direct Measurement Processing

## Configurable Consumption Extracts

Configurable Consumption Extracts are measurement data extracts for a set of usage subscriptions optionally related to a specific market participant.

Configuration options for these extracts include:

- Type of Market Participant, based on Service Point Market Participant or Usage Subscription Market Participant tables and market relationship
- Usage Subscriptions Types associated with the market participant (typically a customer information system receiving bill determinants)
- Unique combinations of unit of measure (UOM), time of use (TOU), and service quantity identifier (SQI) associated with the usage subscriptions
- Specific identifiers used to identify the usage subscriptions, service points, and devices associated with extracted measurement data
- Date and time parameters and frequency of extract
- Extract file naming and compression options

Consumption extracts are exported to a file in a specified file location, either on an application server (in an on-premises implementation) or an Oracle Cloud Infrastructure Object Storage location.

Consumption extracts are created via Consumption Extract Requests. See Consumption Extract Requests in the *Business User Guide*.

This section describes the configuration needed for these types of consumption extracts.

### Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.


Copyright © 2026, Oracle and/or its affiliates.

### Consumption Extract Types

Configuration for consumption extracts are defined via a Consumption Extract Type. See Consumption Extract Type in the *Administrative User Guide* for more information about creating these records.

Configurable consumption extract types should be created using the Extract US Consumption to File (D1-ExtractUSConsumptionToFile) business object. Refer to the embedded help on this business object for more details about the parameters defined for consumption extract types.

Extract US Consumption to File consumption extract types define the following parameters:

- The market participants for which measurement data will be extracted. You define market participants based on their type (Service Point Market Participant or Usage Subscription Market Participant) and market relationship type (retailer, distributor, supplier, and so on), and a specific market participant. Note: If Market Participant Specific is set to "No", the market participant used is based on the Usage Recipient on the Usage Subscription or its Usage Subscription Type.
- The consumption type (interval or scalar)
- One or more Usage Subscription Types and a related unique UOM/TOU/SQI combination
- The specific ID types to include in the extract file for the Usage Subscription, Service Point, child Service Point (if applicable), and Device
- Execution parameters, including:
  - The date and time through which data has been extracted for this consumption extract type
  - A Request Type. Request types used with consumption requests should be created using the Consumption Extract Request Type (D2-IntervalDataExtRepoType) business object, and should reference the Consumption Extract (D2-IntervalDataExtRepository) business object in the **Request Business Object** field. (see The Big Picture of Requests in the *Application Framework Administrative User Guide*)
  - The frequency of extracts for this type (adhoc or daily)
  - Batch controls used for the extract process (see **Batch Controls** below)
- File parameters, including:
  - The file location where extract files will be saved (see External File Storage in the *Application Framework Administrative User Guide*)
  - An optional child folder in the file location
  - File naming options (including an optional prefix and date/time stamp)
  - Whether or not to compress the extract file (file compression is performed using gzip)

### Batch Controls

The data extract process and creation of export files is performed by the following batch controls:

- **Extract Current US Intervals to File** (D1-ECUSI): Used for daily extracts of interval consumption. This process requires that Interval Snapshot Mapping data be generated for any of the interval sizes being extracted. This aides in the efficiency of transposing the interval data into daily interval columns. (see Interval Snapshot Mapping for more information).


Copyright © 2026, Oracle and/or its affiliates.

- **Extract Replacement US Intervals to File** (D1-ERUSI): Used for extracts of replacement historical interval consumption. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table (see **Initial Measurement Subsequent Correction Change Data Capture** below).
- **Extract Current US Scalar Reads to File** (D1-ECUSS): Used for daily extracts of scalar consumption.
- **Extract Replacement US Scalar Reads to File** (D1-ERUSI): Used for extracts of replacement scalar interval consumption. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table (see **Initial Measurement Subsequent Correction Change Data Capture** below).

These batch controls are run for a specific Consumption Extract Type and date range. Batch processing is threaded by on Usage Subscription ID.

Refer to Batch Controls under **Consumption Extract** in the **Oracle Utilities DataConnect** section of the **Integrations** chapter for general information about consumption extract batch processing,

### Initial Measurement Subsequent Correction Change Data Capture

The following algorithms maintain measuring component date/times based on the latest measurement date time for the measuring component. These algorithms also check for a Consumption Extract Type that applies to the current initial measurement's measuring component and identifies whether the initial measurement is for a date/time that has already been extracted.

| Algorithm | Description | Consumption Type |
|---|---|---|
| D1-UPD-DTMC | Update Latest Measurement Date/Time on MC with Consumption Sync | Interval |
| D1-UDTSCMCRE | Update Latest Measurement Date/Time on Scalar MC with Consumption Sync | Scalar |
| D1-UPD-DTMCF | Update Latest Date/Time on MC from a Future Measurement | Interval |
| D1-UPDDTSCMC | Update Latest Date/Time on Scalar Measuring Component | Scalar |

If an initial measurement is for a date/time that has already been extracted, these algorithms create Change Data Capture records in the General Process table that capture the batch code, batch run number, and the initial measurement's ID. The "Replacement" batch processes run off of these records.

Note that the following parameter must be set to “true” on the above algorithms to enable historical change data capture:

- Capture CET Historical Measurement Changes: true
- Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

### Data Extract File Format - Interval

The interval extract process produces JavaScript Object Notation (JSON) files with the extract data described in the following table.

Note that interval extract files can store up to a maximum of 300 intervals. This means that export files can store approximately:


Copyright © 2026, Oracle and/or its affiliates.

- 12.5 days worth of hourly data
- 3 days worth of 15 minute data
- 1 day worth of 5 minute data

| Field | Sample Value | Comments |
|---|---|---|
| usId | 995647003500 | Usage Subscription ID defined by the Consumption Extract Type |
| usType | E-RES | Usage Subscription type |
| pSpId | 458081 | Parent Service Point ID, Null when there is no parent service point. |
| spId | 458081_001 | Child Service Point ID<br>This is the Service Point ID defined by the Consumption Extract Type |
| dvcId | SN823287793219 | Device ID defined by the Consumption Extract Type |
| uomTouSqi | KWH// | UOM / TOU / SQI, separated by forward slashes ("/") |
| tz | US/Eastern | Time zone of the service point.<br>This is the time zone name from the CI_TIME_ZONE table not the time zone code. |
| intPerDay | 288 | Number of Intervals for the day, will be 1 hour less or 1 hour more of intervals for DST transitions |
| intSize | 300 | Length of each interval in seconds |
| mktPart | DIRECT_ENERGY | Market Participant ID of the market participant (if applicable) |
| stDttm | 2020-01-01T00:00:00-05:00 | Start of the day, similar to initial measurement, this represents the start of the first interval. |
| q1 | 1.325 | Measurement Value for the first interval, the measurement date/time for this interval would be the stDttm plus the intSize |
| c1 | 301000 | Measurement Condition for the first interval.<br>Null when "Regular" (501000) |
| q2 | 1.230 | |
| c2 | 301000 | |
| ... | q/c 3 through 299 here | |
| q300 | .9134 | |
| c300 | 30100 | |

**Example:**

```
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "intPerDay":"288", "intSize":"300",
"mktPart":"DIRECT_ENERGY",
 "stDttm":"2020-01-01T00:00:00-05:00", "q1":"1.325", "c1":"301000",
```


Copyright © 2026, Oracle and/or its affiliates.

```
"q2":"1.325",
 "c2":"301000" "q3":"1.325", "c3":"301000",
<intervening intervals>, "q300":"1.325", "c300":"301000"}
```

### Data Extract File Format - Scalar

The scalar extract process produces JavaScript Object Notation (JSON) files with the extract data described in the following table.

| Name | Sample Value | Comments |
|---|---|---|
| c | 301000 | Measurement Condition for the scalar reading<br>Null when "Regular" (501000) |
| dvcId | SN823287793219 | Device ID defined by the Consumption Extract Type |
| enDttm | 2020-01-02T00:00:00-05:00 | End date/time of the reading |
| mktPart | DIRECT_ENERGY | Market Participant ID of the market participant (if applicable) |
| pSpId | 458081 | |
| q | 1.325 | Measurement Value for scalar reading |
| r | 678645 | Dial reading of the meter |
| spId | 458081_001 | Service Point ID defined by the Consumption Extract Type |
| stDttm | 2020-01-01T00:00:00-05:00 | Start date/time of the reading (the End Date/Time of the previous reading) |
| tz | US/Eastern | Time zone of the service point.<br>This is the time zone name from the CI_TIME_ZONE table not the time zone code. |
| uomTouSqi | KWH// | UOM / TOU / SQI, separated by forward slashes ("/") |
| usId | 995647003500 | Usage Subscription ID defined by the Consumption Extract Type |
| usType | E-RES | Usage subscription type |

**Example (3 records):**

```
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
```


Copyright © 2026, Oracle and/or its affiliates.

```
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
{"usId":"995647003500", "usType":"E-RES", "pSpId":"458081",
"spId":"458081_001", "dvcId":"SN823287793219", "uomTouSqi":"KWH//",
 "tz":"US/Eastern", "mktPart":"DIRECT_ENERGY",
"stDttm":"2020-01-01T00:00:00-05:00",
 "enDttm":"2020-01-02T00:00:00-05:00", "q":"1.325", "c":"301000",
"r":"678645"}
```


### Extending the Extract Processing

Some implementations may wish to extend and/or customize the extract process based on specific requirements.

#### Extending List of IDs To Use

Consumption Extract Types define specific types of IDs to use for Usage Subscriptions, Service Points, child Service Points (if applicable), and Devices. If an implementation wishes to extend the types of IDs available, they need to add appropriate values to the lookup flags used by each of ID type. The table below lists the lookup codes used by the "IDs to Use" fields.

| ID Type | Lookup Flag |
|---|---|
| Usage Subscription ID | Usage Subscription Identifier Type (US_ID_TYPE_FLG) |
| Service Point ID (used for both parent and child service points) | Service Point Identifier Type (SP_ID_TYPE_FLG) |
| Device ID | Device Identifier Type (DVC_ID_TYPE_FLG) |

#### Processing Records and Creating Output Files

The extract process leverages the generic Plug-in Driven Background Processes functionality but works in a slightly different manner. Instead of using the Select Records algorithm to retrieve the IDs of the individual records to be processed, that algorithm returns a range of Usage Subscription IDs to be extracted and the Process Records algorithm performs the detailed query to extract the range of Usage Subscription IDs.

This provides two benefits:

1. There is no limitation on the length of the query in the Process Records algorithm.
2. Retrieving data for many Usage Subscriptions at a single time is significantly more efficient.

This means that any required customization can be achieved by creating a new custom (CM) batch control and duplicating the Process Records Groovy script. Changes can be made within the custom Groovy script to the extraction SQL to retrieve additional information or provide additional selection criteria. If any changes are needed to the file format that can be done within the record processing logic that populates the file.



Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> The Process Records scripts include a number of steps each containing specific processing logic and SQL based on the configuration of the Market Participants on the Consumption Extract Type as well as whether or not the specified UOMs measure peak quantities or not. Be sure to edit each of these steps as appropriate when creating customized versions of these scripts.

# Maintenance Object Data Export

This section describes the process by which data can be exported from maintenance objects used with Oracle Utilities Customer Cloud Service and Oracle Utilities Meter Solution Cloud Service for use in external applications and systems.

This section provides details regarding how the data export process is used with specific maintenance objects. See Data Export in the *Application Framework Administrative User Guide* for a general description of the data export process.

- Enabling Maintenance Objects for Generalized Data Export
- Measurement Data Export
- Initial Measurement Data Header Export
- Aggregated Measurement Data Export

## Enabling Maintenance Objects for Generalized Data Export

Export of data from most maintenance objects is supported via "Generalized" export. See Generalized Data Export in the *Application Framework Administrative User Guide* for more information about generalized data export.

### Data Export Control Records for Maintenance Objects

Data export from a maintenance object requires a Data Export Control record configured for that maintenance object. See **Enabling Data Export** in Generalized Data Export in the *Application Framework Administrative User Guide* for more information.

### Change Data Capture for Measuring Components and Usage Subscriptions

For most maintenance objects, changes are tracked via the Capture Change for Ongoing Export audit algorithm (F1-MO-REGCHN).

Measuring Components and Usage Subscriptions use a "Most Recent" date field to track related transactions (Initial Measurements and Measurements in the case of Measuring Components and Usage Transactions in the case of Usage Subscriptions). These "Most Recent" fields are updated with each new initial measurement or usage transaction created by the system.

In order to prevent the Change Data Capture process from capturing changes based on these frequent transactions (daily for measurements and monthly for usage transactions), the Capture Change for Ongoing Export audit algorithm has been disabled on the Measuring Component and Usage Subscription maintenance objects.

Enabling change data capture and generalized data export for these maintenance objects requires adding the “Generalized Data Export BO Change Data Capture” (D1-BO-REGCHN)


Copyright © 2026, Oracle and/or its affiliates.

Audit algorithm to the appropriate Measuring Component and Usage Subscription business objects. This algorithm creates records in the Data Export Update table used by the generalized export process.

## Measurement Data Export

Export of data from the Measurement maintenance object is supported via "Specialized" export. See **Specialized Data Export** in the *Application Framework Administrative User Guide* for more information about specialized data export.

The Measurement maintenance object is configured to support data export via the following options:

| Option Type | Option Value |
|---|---|
| Data Export Class | F1SE (Specialized) |
| Export Batch Control | D1-SMDEC (Specialized Measurement Data Extract Current) |

### Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

### Initial vs. Current vs. Historical Extract

There are three types of extracts used with the Measurement maintenance object:

- **Initial Extracts** gather an initial extract of measurement data, controlled by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of the measurement data prior to using the daily current and daily historical extracts.
- **Daily Current Extracts** gather measurement data for measurements within the last 30 days that have been either added or changed. This extract is based on the **Measurement Date/Time** and **Last Update Timestamp** fields on the Measurement table, as well as the **Export Through Date/Time** field on the Data Export Control record (see below) to include only changes since the last extract.
- **Daily Historical Extracts** gather measurement data for measurements older than the last 30 days that have either been added or changed. To avoid having to scan the entire Measurement table, this extract is based on Change Data Capture. During initial measurement data processing if an initial measurement has measurements that are older than 30 days, then a Change Data Capture record is created in the General Process or Measurement Extract Request table.
  Enabling historical change data capture requires configuration of the following algorithm parameters:
  - Capture Historical Measurement Changes: true
  - Historical Measurement Data Extract Batch Control: D1-SMDEH (Specialized Measurement Data Extract Historical)
  - Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

  These parameters should be configured as described on the following algorithms:


Copyright © 2026, Oracle and/or its affiliates.

- Update Latest Measurement Date/Time on MC with Consumption Sync (D1-UPD-DTMC)
- Update Latest Measurement Date/Time on Scalar MC with Consumption Sync (D1-UDTSCMCRE)
- Update Latest Date/Time on MC from a Future Measurement (D1-UPD-DTMCF)
- Update Latest Date/Time on Scalar Measuring Component (D1-UPDDTSCMC)
- Re-derive other values (D1-REDEROVAL)

> **Note**
>
> This use of Change Data Capture is an exception to the general rule described in the **Specialized Data Export** section of the *Application Framework Administrative User Guide*.

### Data Export Control Records

Data export from the Measurement maintenance object requires a Data Export Control record. This record designates that the Measurement maintenance object should be exported, and tracks the date and time through which data has been exported so far, allowing subsequent export processes to only export incremental changes.

See **Defining Data Export Control** in the *Application Framework Administrative User Guide* for more information about creating Data Export Control records.

### Batch Controls

Export of data and creation of export files is performed by the following batch controls:

- **Specialized Measurement Initial Extract** (D1-SMDIE): Used for initial extract of measurement data. Unlike the current extract this batch is controlled directly by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of the measurement data prior to using the current extract (D1-SMDEC) and the historical extract (D1-SMDEH). For best performance it is recommended to break up the initial extract period into multiple smaller periods and to use multiple threads for each period. The extract will retrieve all measurements with a measurement date/time that is greater than the input start date/time and less than or equal to the input end date/time.
- **Specialized Measurement Data Extract Current** (D1-SMDEC): Used for daily extracts of current (new) measurement data. This process extracts data that has changed since the last extract process has run (based on the **Export Through Date/Time** field on the Data Export Control record. This process updates the **Export Through Date/Time** field at the start of batch processing to eliminate the possibility of not setting an initial measurement as historical by Change Data Capture during the processing window.
- **Specialized Measurement Data Extract Historical** (D1-SMDEH): Used for daily extracts of historical measurement data. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table.

Batch processing is threaded by on Measuring Component ID, and creates one or more files based on the number of threads processed and batch control parameters.

### File Creation and Naming

The extract process uses two controls to determine the number of files to be created:


Copyright © 2026, Oracle and/or its affiliates.

1. Each thread will create at least 1 file (more threads result in smaller files).
2. The **File Size Limit** parameter can be used to specify set the number of measurement rows to include in any given file. This can result in a single thread producing multiple files.

The data in each file is not ordered, each thread will retrieve measurement data for a set of Measuring Component IDs, but beyond that there will be no control for how the data is loaded into the file.

**File Naming**

Since each thread can produce multiple files and the number of files per thread cannot be predicted the file names will also include an indicator for the first file in a thread and the last file in a thread.

- MSRMT_EXPORT_<file prefix batch parameter>_<batch number>_<batch thread number>_<total thread count>_<thread file indicator>_<process date time>.csv.gz

where:

- <file prefix batch parameter> is an optional file prefix defined using the **File Name Prefix** batch control parameter
- <batch number> is the batch number for the process
- <batch thread number> is the thread number that created the file
- <total thread count> is the total number of threads
- <thread file indicator>: is an identified for the first and last files created by a thread, S for first file E for last file
- <process date time>is the date and time at which the file was created, in YYYYMMDDhhmmss format (YYYY = year, MM = month, DD = day, hh = hour, mm = minute, ss = second)

Examples: For batch number 135, comprising 10 threads:

- First file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_S_20191104173233.csv.gz
- All files before the last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_20191104173233.csv.gz
- Last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_E_20191104173233.csv.gz

**File Format**

The extract process produces comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column are included in the output file as a header row if the **Include Header** parameter on the batch controls is set to “Y”.

> **Note**
>
> All date/times will be converted from OUAF format to the appropriate ISO format in UTC with the time zone provided (as "Z").


Copyright © 2026, Oracle and/or its affiliates.

| | | | | Applicability | | |
|---|---|---|---|---|---|---|
| **Field** | **Data Type** | **Sample Content** | **Description** | **Interval** | **Scalar** | **Subtractive Interval** |
| MEASR_COMP_ID | CHAR(12) | 000004769213 | Primary Key of Channel from Meter (aka Device) | X | X | X |
| MSRMT_DTTM | DATE | 2019-01-01T00:00:00Z | End of the time period the measurement covers. For example, if this were an hourly interval it would cover 2018-12-31T23:00:00Z to 2019-01-01T00:00:00Z. | X | X | X |
| PREV_MSRMT_DTTM | DATE | 2018-12-31T00:00:00Z | For scalar measurements this identifies the start of the period being measured. Note: there should be another measurement for the same MC with this date as the MSRMT_DTTM. | | X | |
| MSRMT_COND_FLG | CHAR(6) | 501000 | Defines the quality of the measurement (e.g. estimated, regular, super).<br>Valid values defined on F1_EXTENDABLE_LOOKUP_VAL for the BUS_OBJ_CD D1-MeasurementConditionLookup. | X | X | X |
| MSRMT_USE_FLG | CHAR(4) | D101 | Indicates whether the measurement is usable.<br>Valid values defined on CI_LOOKUP. | | X | |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | Applicability | | |
|---|---|---|---|---|---|---|
| **Field** | **Data Type** | **Sample Content** | **Description** | **Interval** | **Scalar** | **Subtractive Interval** |
| USER_EDITED_FLG | CHAR(4) | D1UE | Indicates whether the user has edited the measurement.<br>Valid values defined on CI_LOOKUP. | X | X | X |
| READING_VAL | NUMBER(16,6) | 750000000 | This is the Dial Reading which is used to calculate the consumed amount. | | X | X |
| READING_COND_FLG | CHAR(6) | 501000 | Defines the quality of the Daily Reading (e.g. estimated, regular, super).<br>Valid values defined on F1_EXTENDABLE_LOOKUP_VAL for the BUS_OBJ_CD D1-MeasurementConditionLookup. | | | X |
| COMBINED_MULTIPLIER | NUMBER(18,6) | 1 | Contains the product of the Measuring Component Multiplier and the Installation Constant. The MSRMT_VAL will have been multiplied by this number. | X | X | X |
| MSRMT_VAL | NUMBER(16,6) | 1.23 | This is the consumption reported from the meter after going through VEE (aka billing ready). | X | X | X |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | Applicability | | |
|---|---|---|---|---|---|---|
| **Field** | **Data Type** | **Sample Content** | **Description** | **Interval** | **Scalar** | **Subtractive Interval** |
| MSRMT_VAL1 | NUMBER(16,6) | | This and the subsequent measurement values (MSRMT_VAL 1 through 10) are derived values that are computed from the core MSMRT_VAL.<br>Examples of this are UOM conversion (CCF to THERM) or loss adjustment. | X | X | X |
| MSRMT_VAL2 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL3 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL4 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL5 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL6 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL7 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL8 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL9 | NUMBER(16,6) | | | X | X | X |
| MSRMT_VAL10 | NUMBER(16,6) | | | X | X | X |
| CRE_DTTM | DATE | 2019-10-31T11:09:31Z | When the record was created. | X | X | X |
| LAST_UPDATE_DTTM | DATE | 2019-11-09T16:35:02Z | When the record was last updated. | X | X | X |

### Sample Export File

The following is a sample export file for interval data spanning a six hour window. Note that this does not include labels or column headers, but the data is in the same order as the column descriptions above in the **File Format** section above.

```
000004769213,2019-01-01T08:00:00Z,,501000,,,,1,1.3715,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T09:00:00Z,,501000,,,,1,1.882,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T10:00:00Z,,501000,,,,1,1.838,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T11:00:00Z,,501000,,,,1,1.8585,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T12:00:00Z,,501000,,,,1,1.836,0,0,0,0,0,0,0,0,0,0,2019
```


Copyright © 2026, Oracle and/or its affiliates.

```
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T13:00:00Z,,501000,,,,1,1.83,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:39:09Z,2019-08-09T20:39:09Z
000671510494,2019-01-01T08:00:00Z,,501000,,,,1,0.486,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T09:00:00Z,,501000,,,,1,0.4965,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T10:00:00Z,,501000,,,,1,0.45,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T11:00:00Z,,501000,,,,1,0.4725,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T12:00:00Z,,501000,,,,1,0.4575,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T13:00:00Z,,501000,,,,1,0.447,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
020643301975,2019-01-01T08:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T09:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T10:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T11:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T12:00:00Z,,501000,,,,1,0.0165,0,0,0,0,0,0,0,0,0,0,201
9-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T13:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
```

## Initial Measurement Data Header Export

Export of header data from the Initial Measurement Data maintenance object is supported via "Specialized" export. See **Specialized Data Export** in the *Application Framework Administrative User Guide* for more information about specialized data export.

The Initial Measurement Data maintenance object is configured to support data export via the following options:

| Option Type | Option Value |
|---|---|
| Data Export Class | F1SE (Specialized) |
| Export Batch Control | D1-SIMDC (Specialized IMD Header Extract Current) |

> **Note**
>
> When Initial Measurement Data Header Export is used with Direct Measurement Processing, exports contain only IMD header information for exceptions, including Informational Exception Headers created due to "Informational" exceptions.

### Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order


Copyright © 2026, Oracle and/or its affiliates.

to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

### Initial vs. Current vs. Historical Extract

There are three types of extracts used with the Initial Measurement Data maintenance object:

- **Initial Extracts** gather an initial extract of initial measurement data headers, controlled by providing a start and end date time of the initial measurement data headers that should be extracted. This process should be used to perform the initial extract of the data prior to using the daily current and daily historical extracts.
- **Daily Current Extracts** gather measurement data for measurements within the last 30 days that have been either added or changed. This extract is based on the **Last Update Timestamp** field on the Initial Measurement Data table, as well as the **Export Through Date/Time** field on the Data Export Control record (see below) to include only changes since the last extract.
- **Daily Historical Extracts** gather data for initial measurements older than the last 30 days that have either been added or changed. To avoid having to scan the entire Initial Measurement Data table, this extract is based on Change Data Capture. During initial measurement data processing if an initial measurement has measurements that are older than 30 days, then a Change Data Capture record is created in the General Process or Measurement Extract Request table.
  Enabling historical change data capture requires configuration of the following algorithm parameters:
  - Capture Historical Measurement Changes: true
  - Historical IMD Extract Batch Control: D1-SMDEH (Specialized IMD Header Extract - Historical)
  - Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

  These parameters should be configured as described on the following algorithms:
  - Update Latest Measurement Date/Time on MC with Consumption Sync (D1-UPD-DTMC)
  - Update Latest Measurement Date/Time on Scalar MC with Consumption Sync (D1-UDTSCMCRE)
  - Update Latest Date/Time on MC from a Future Measurement (D1-UPD-DTMCF)
  - Update Latest Date/Time on Scalar Measuring Component (D1-UPDDTSCMC)

> **Note**
>
> This use of Change Data Capture is an exception to the general rule described in the **Specialized Data Export** section of the *Application Framework Administrative User Guide*.

### Data Export Control Records for Maintenance Objects

Data export from the Initial Measurement Data maintenance object requires a Data Export Control record. This record designates that the Initial Measurement Data maintenance object should be exported, and tracks the date and time through which data has been exported so far, allowing subsequent export processes to only export incremental changes.


Copyright © 2026, Oracle and/or its affiliates.

See **Defining Data Export Control** in the *Application Framework Administrative User Guide* for more information about creating Data Export Control records.

### Batch Controls

Export of data and creation of export files is performed by the following batch controls:

- **Specialized IMD Header Initial Extract** (D1-SIMDI): Used for initial extract of initial measurement data. Unlike the current extract this batch is controlled directly by providing a start and end date time of the IMD headers that should be extracted. This process should be used to perform the initial extract of the IMD header data prior to using the current extract (D1-SIMDC) and the historical extract (D1-SIMDH). For best performance it is recommended to break up the initial extract period into multiple smaller periods and to use multiple threads for each period. The extract will retrieve all IMD headers with a IMD to date/time that is greater than the input start date/time and less than or equal to the input end date/time.
- **Specialized IMD Header Data Extract - Current** (D1-SIMDC): Used for daily extracts of current (new) initial measurement data. This process extracts data that has changed since the last extract process has run (based on **the Export Through Date/Time** field on the Data Export Control record). This process updates the **Export Through Date/Time** field at the start of batch processing to eliminate the possibility of not setting an initial measurement as historical by Change Data Capture during the processing window.
- **Specialized IMD Header Extract - Historical** (D1-SIMDH): Used for daily extracts of historical initial measurement data. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table.

Batch processing is threaded by on Initial Measurement Data ID, and creates one or more files based on the number of threads processed and batch control parameters.

### File Creation and Naming

The extract process uses two controls to determine the number of files to be created:

1. Each thread will create at least 1 file (more threads result in smaller files)
2. The **File Size Limit** parameter can be used to specify set the number of initial measurement data rows to include in any given file. This can result in a single thread producing multiple files.

The data in each file is not ordered, each thread will retrieve initial measurement data for a set of Initial Measurement Data IDs, but beyond that there will be no control for how the data is loaded into the file.

### File Naming

Since each thread can produce multiple files and the number of files per thread cannot be predicted the file names will also include an indicator for the first file in a thread and the last file in a thread.

- IMD_EXPORT_<file prefix batch parameter>_<batch number>_<batch thread number>_<total thread count>_<file number>_<last file indicator>_<process date>.csv.gz

where:

- <file prefix batch parameter> is an optional file prefix defined using the **File Name Prefix** batch control parameter
- <batch number> is the batch number for the process
- <batch thread number> is the thread number that created the file


Copyright © 2026, Oracle and/or its affiliates.

- <total thread count> is the total number of threads
- <file number> is the number of the file
- <last file indicator>: identifies the last files created by a thread (E)
- <process date time>is the date and time at which the file was created, in YYYYMMDDhhmmss format (YYYY = year, MM = month, DD = day, hh = hour, mm = minute, ss = second)

Examples: For batch number 135, comprising 10 threads:

- First file of 3rd out of 10 threads: IMD_EXPORT_SOME_PREFIX_135_3_10_1_20191104173233.csv.gz
- All files before the last file of 3rd out of 10 threads: IMD_EXPORT_SOME_PREFIX_135_3_10_2_20191104173233.csv.gz
- Last file of 3rd out of 10 threads: IMD_EXPORT_SOME_PREFIX_135_3_10_E_20191104173233.csv.gz

### File Format

The extract process produces comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column are included in the output file as a header row if the **Include Header** parameter on the batch controls is set to “Y”.

> **Note**
>
> All date/times will be converted from OUAF format to the appropriate ISO format in UTC with the time zone provided (as "Z").

| | | | | Applicability | | |
|---|---|---|---|---|---|---|
| **Field** | **Data Type** | **Sample Content** | **Description** | **Interval** | **Scalar** | **Subtractive Interval** |
| INIT_MSRMT_DATA_ID | CHAR(12) | 524822451166 04 | Primary Key of the initial measurement | X | X | X |
| MEASR_COMP_ID | CHAR(12) | 000004769213 | Primary Key of the measuring component for the initial measurement | X | X | X |
| IMD_EXT_ID | CHAR(12) | external_file | External ID for the initial measurement. Usually the name of the file that the initial measurement data was received in. | X | X | X |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | Applicability | | |
|---|---|---|---|---|---|---|
| **Field** | **Data Type** | **Sample Content** | **Description** | **Interval** | **Scalar** | **Subtractive Interval** |
| BUS_OBJ_CD | CHAR(12) | D1-ManualIMDInterval | The business object that governs the initial measurement | X | X | X |
| BO_STATUS_CD | VARCHAR(30) | COMPLETED | The status of the initial measurement | X | X | X |
| BO_STATUS_REASON_CD | VARCHAR(30) | FORCE COMPLETE | A reason code for the status. Generally populated when there was a VEE exception that was overridden and the initial measurement was completed without fixing the error. | X | X | X |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | Applicability | | |
|---|---|---|---|---|---|---|
| **Field** | **Data Type** | **Sample Content** | **Description** | **Interval** | **Scalar** | **Subtractive Interval** |
| D1_FROM_DTTM | DATE/TIME | 2019-01-01T00:00:00Z | Start of the time period the initial measurement covers.<br>For subtractive scalar data this represents the prior measurement that will be used as the start reading when calculating consumption.<br>For interval data this represents the start of the first interval for the initial measurement. Note: no measurements will be created for this date/time as MDM represents intervals as the end of the interval. So the first interval of the initial measurement will be this date/time plus the interval size (e.g. if the interval size is 15 minutes then the first interval would be 15 minutes after this date/time). | X | X | X |


Copyright © 2026, Oracle and/or its affiliates.

| | | | | Applicability | | |
|---|---|---|---|---|---|---|
| **Field** | **Data Type** | **Sample Content** | **Description** | **Interval** | **Scalar** | **Subtractive Interval** |
| D1_TO_DTTM | DATE/TIME | 2019-01-02T00:00:00Z | End of the time period the initial measurement covers.<br>For scalar data this date/time will match the measurement.<br>For interval data this will match the date/time of the last interval measurement. | X | X | X |
| CRE_DTTM | DATE/TIME | 2019-01-03T00:00:00Z | Date and time when the initial measurement was initially created | X | X | X |
| STATUS_UPD_DTTM | DATE/TIME | 2019-01-03T00:00:00Z | Date and time when the initial measurement's status was last updated | X | X | X |
| LAST_UPDATE_DTTM | DATE/TIME | 2019-01-03T00:00:00Z | Date and time when the initial measurement was last updated.<br>Note: updates can happen outside of a status change so this can be more recent than the STATUS_UPD_DTTM. | X | X | X |

### Sample Export File

The following is a sample export file containing initial measurement header records. Note that this does not include labels or column headers, but the data is in the same order as the column descriptions above in the **File Format** section above.

```
55394295586154,553275345711,external file1,D1-
ManualIMDScalar,ERROR,,2020-01-02T08:00:00Z,2022-01-01T08:00:00Z,2020-08-17T18
:47:41Z,2020-08-17T18:47:41Z,2020-08-17T18:48:54Z
```


Copyright © 2026, Oracle and/or its affiliates.

```
55392270285377,553886220134,external file2,D1-
ManualIMDInterval,ERROR,,2022-01-01T08:00:00Z,2022-01-02T08:00:00Z,2020-08-15T
04:18:43Z,2020-08-15T04:25:57Z,2020-08-15T04:25:57Z
55383485376359,553886220134,external file3,D1-
ManualIMDInterval,ERROR,,2022-01-01T08:00:00Z,2022-01-02T08:00:00Z,2020-08-15T
04:47:36Z,2020-08-15T04:55:03Z,2020-08-15T04:55:03Z
```

# Aggregated Measurement Data Export

Export of data from the Aggregated Measurement maintenance object (D1-AGGMSRMT) is supported via "Specialized" export. See **Specialized Data Export** in the *Application Framework Administrative User Guide* for more information about specialized data export. The Aggregated Measurement maintenance object is configured to support data export via the following options:

| Option Type | Option Value |
|---|---|
| Data Export Class | F1SE (Specialized) |
| Export Batch Control | D1-SAMDC (Specialized Agg. Measurement Extract Cur.) |

### Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

### Initial vs. Current vs. Historical Extract

There are three types of extracts used with the Aggregated Measurement maintenance object:

- **Initial Extracts** gather an initial extract of aggregated measurement data, controlled by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of the aggregated measurement data prior to using the daily current and daily historical extracts.
- **Daily Current Extracts** gather aggregated measurement data for measurements within the last 30 days that have been either added or changed. This extract is based on the Measurement Date/Time and Last Update Timestamp fields on the Aggregated Measurement table, as well as the Export Through Date/Time field on the Data Export Control record (see below) to include only changes since the last extract.
- **Daily Historical Extracts** gather aggregated measurement data for measurements older than the last 30 days that have either been added or changed. To avoid having to scan the entire Aggregated Measurement table, this extract is based on Change Data Capture. During initial measurement data processing if an aggregated measurement has measurements that are older than 30 days, then a Change Data Capture record is created in the General Process or Measurement Extract Request table.

Enabling historical change data capture for aggregated measurement data requires configuration of the following algorithm parameters:

- Capture Historical Aggregation Changes: true
- Aggregation Historical Batch Control: D1-SAMDH (Specialized Aggregated Measurement Data Extract Historical)


Copyright © 2026, Oracle and/or its affiliates.

- Capture Historical Measurement Changes: true (when using Direct Measurement Processing)

These parameters should be configured as described on the following algorithms:

- Create General Process Record for Historical Dynamic Aggregation (D1CGPHISAGG)
- Create General Process Record for Historical Static Aggregation (D1CGPHISA)

> **Note**
>
> This use of Change Data Capture is an exception to the general rule described in the **Specialized Data Export** section of the *Application Framework Administrative User Guide*.

#### Data Export Control Records

Data export from the Aggregated Measurement maintenance object requires a Data Export Control record. This record designates that the Aggregated Measurement maintenance object should be exported, and tracks the date and time through which data has been exported so far, allowing subsequent export processes to only export incremental changes.

See **Defining Data Export Control** in the *Application Framework Administrative User Guide* for more information about creating Data Export Control records.

#### Batch Controls

Export of data and creation of export files is performed by the following batch controls:

- **Specialized Aggregate Msrmt. Initial** (D1-SAMDI): Used for initial extract of aggregated measurement data. Unlike the current extract this batch is controlled directly by providing a start and end date time of the measurements that should be extracted. This process should be used to perform the initial extract of aggregated measurement data prior to using the current extract (D1-SAMDC) and the historical extract (D1-SAMDH). For best performance it is recommended to break up the initial extract period into multiple smaller periods and to use multiple threads for each period. The extract will retrieve all aggregated measurements with a measurement date/time that is greater than the input start date/time and less than or equal to the input end date/time.
- **Specialized Agg. Measurement Extract Cur.** (D1-SAMDC): Used for daily extracts of current (new) aggregated measurement data. This process extracts data that has changed since the last extract process has run (based on the **Export Through Date/Time** field on the Data Export Control record. This process updates the **Export Through Date/Time** field at the start of batch processing to eliminate the possibility of not setting an initial measurement as historical by Change Data Capture during the processing window.
- **Specialized Aggregated Measurement Data Extract Historical** (D1-SAMDH): Used for daily extracts of historical aggregated measurement data. This process runs against Change Data Capture records in the General Process or Measurement Extract Request table

Batch processing is threaded by on Measuring Component ID, and creates one or more files based on the number of threads processed and batch control parameters.

#### File Creation and Naming

The extract process uses two controls to determine the number of files to be created:

1. Each thread will create at least 1 file (more threads result in smaller files).


Copyright © 2026, Oracle and/or its affiliates.

2. The **File Size Limit** parameter can be used to specify set the number of aggregated measurement rows to include in any given file. This can result in a single thread producing multiple files.

The data in each file is not ordered, each thread will retrieve aggregated measurement data for a set of Measuring Component IDs, but beyond that there will be no control for how the data is loaded into the file.

**File Naming**

Since each thread can produce multiple files and the number of files per thread cannot be predicted the file names will also include an indicator for the first file in a thread and the last file in a thread.

- MSRMT_EXPORT_<file prefix batch parameter>_<batch number>_<batch thread number>_<total thread count>_<thread file indicator>_<process date time>.csv.gz

where:

- <file prefix batch parameter> is an optional file prefix defined using the **File Name Prefix** batch control parameter
- <batch number> is the batch number for the process
- <batch thread number> is the thread number that created the file
- <total thread count> is the total number of threads
- <thread file indicator>: is an identified for the first and last files created by a thread, S for first file E for last file
- <process date time>is the date and time at which the file was created, in YYYYMMDDhhmmss format (YYYY = year, MM = month, DD = day, hh = hour, mm = minute, ss = second)

Examples: For batch number 135, comprising 10 threads:

- First file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_S_20191104173233.csv.gz
- All files before the last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_20191104173233.csv.gz
- Last file of 3rd out of 10 threads: MSRMT_EXPORT_PREFIX_135_3_10_E_20191104173233.csv.gz

**File Format**

The extract process produces comma separated values (CSV) files with the data described in the following table. The field names in the **Field** column are included in the output file as a header row if the **Include Header** parameter on the batch controls is set to “Y”.

> **Note**
>
> All date/times will be converted from OUAF format to the appropriate ISO format in UTC with the time zone provided (as "Z").

| Field | Data Type | Sample Content | Description |
|---|---|---|---|
| MEASR_COMP_ID | CHAR(12) | 000004769213 | Primary Key of Channel from Meter (aka Device) |


Copyright © 2026, Oracle and/or its affiliates.

| Field | Data Type | Sample Content | Description |
|---|---|---|---|
| MSRMT_DTTM | DATE | 2019-01-01T00:00:00Z | End of the time period the measurement covers. For example, if this were an hourly interval it would cover 2018-12-31T23:00:00Z to 2019-01-01T00:00:00Z. |
| PREV_MSRMT_DTTM | DATE | 2018-12-31T00:00:00Z | For scalar measurements this identifies the start of the period being measured. Note: there should be another measurement for the same MC with this date as the MSRMT_DTTM. |
| MSRMT_COND_FLG | CHAR(6) | 501000 | Defines the quality of the measurement (e.g. estimated, regular, super). Valid values defined on F1_EXTENDABLE_LOOKUP_VAL for the BUS_OBJ_CD D1-MeasurementConditionLookup. |
| MSRMT_USE_FLG | CHAR(4) | D101 | Indicates whether the measurement is usable. Valid values defined on CI_LOOKUP. |
| USER_EDITED_FLG | CHAR(4) | D1UE | Indicates whether the user has edited the measurement. Valid values defined on CI_LOOKUP. |
| READING_VAL | | | Values for this field are not included in the aggregated measurement export. |
| READING_COND_FLG | | | Values for this field are not included in the aggregated measurement export. |
| COMBINED_MULTIPLIER | | | Values for this field are not included in the aggregated measurement export. |
| MSRMT_VAL | NUMBER(16,6) | 1.23 | This is the consumption reported from the meter after going through VEE (aka billing ready). |


Copyright © 2026, Oracle and/or its affiliates.

| Field | Data Type | Sample Content | Description |
|---|---|---|---|
| MSRMT_VAL1 | NUMBER(16,6) | | This and the subsequent measurement values (MSRMT_VAL 1 through 10) are derived values that are computed from the core MSMRT_VAL. Examples of this are UOM conversion (CCF to THERM) or loss adjustment. |
| MSRMT_VAL2 | NUMBER(16,6) | | |
| MSRMT_VAL3 | NUMBER(16,6) | | |
| MSRMT_VAL4 | NUMBER(16,6) | | |
| MSRMT_VAL5 | NUMBER(16,6) | | |
| MSRMT_VAL6 | NUMBER(16,6) | | |
| MSRMT_VAL7 | NUMBER(16,6) | | |
| MSRMT_VAL8 | NUMBER(16,6) | | |
| MSRMT_VAL9 | NUMBER(16,6) | | |
| MSRMT_VAL10 | NUMBER(16,6) | | |
| CRE_DTTM | DATE | 2019-10-31T11:09:31Z | When the record was created. |
| LAST_UPDATE_DTTM | DATE | 2019-11-09T16:35:02Z | When the record was last updated. |

### Sample File

The following is a sample export file for interval data spanning a six hour window. Note that this does not include labels or column headers, but the data is in the same order as the column descriptions above in the **File Format** section above.

```
000004769213,2019-01-01T08:00:00Z,,501000,,,,,1,1.3715,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T09:00:00Z,,501000,,,,,1,1.882,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T10:00:00Z,,501000,,,,,1,1.838,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T11:00:00Z,,501000,,,,,1,1.8585,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T12:00:00Z,,501000,,,,,1,1.836,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:39:09Z,2019-08-09T20:39:09Z
000004769213,2019-01-01T13:00:00Z,,501000,,,,,1,1.83,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:39:09Z,2019-08-09T20:39:09Z
000671510494,2019-01-01T08:00:00Z,,501000,,,,,1,0.486,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T09:00:00Z,,501000,,,,,1,0.4965,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T10:00:00Z,,501000,,,,,1,0.45,0,0,0,0,0,0,0,0,0,0,2019-
08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T11:00:00Z,,501000,,,,,1,0.4725,0,0,0,0,0,0,0,0,0,0,201
```


Copyright © 2026, Oracle and/or its affiliates.

```
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T12:00:00Z,,501000,,,,1,0.4575,0,0,0,0,0,0,0,0,0,0,201
9-08-09T20:56:47Z,2019-08-09T20:56:47Z
000671510494,2019-01-01T13:00:00Z,,501000,,,,1,0.447,0,0,0,0,0,0,0,0,0,0,2019
-08-09T20:56:47Z,2019-08-09T20:56:47Z
020643301975,2019-01-01T08:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T09:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T10:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T11:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T12:00:00Z,,501000,,,,1,0.0165,0,0,0,0,0,0,0,0,0,0,201
9-08-09T21:04:28Z,2019-08-09T21:04:28Z
020643301975,2019-01-01T13:00:00Z,,501000,,,,1,0.015,0,0,0,0,0,0,0,0,0,0,2019
-08-09T21:04:28Z,2019-08-09T21:04:28Z
```

# Service Point Consumption Extracts

Service Point Consumption extracts are measurement data extracts for a set of service points assigned to an entity group.

Configuration options for these extracts include:

- Type of measurement data to extract (interval or scalar)
- Unique combinations of unit of measure (UOM), time of use (TOU), and service quantity identifier (SQI) associated with the usage subscriptions
- Date and time parameters and frequency of extract
- Extract file naming and location

Service Point Consumption extracts are created via Requests. Service Point Consumption extracts are exported to a file in a specified file location, either on an application server (in an on-premises implementation) or an Oracle Cloud Infrastructure Object Storage location. External URL attachments linking to extract files are created on the Request for each extract.

This section describes the configuration needed for these types of extracts.

> **Note**
>
> Service Point Consumption Extracts are only available in cloud (SaaS) implementations.

### Service Point Consumption Extract Entity Groups

Entity groups used with Service Point Consumption Extracts must be created using the Service Point Measurement Extract Group (D1-SPMeasurementExtractGroup) business object. These entity groups will have a Group Category of "SP Measurement Extract".

Refer to Understanding Entity Groups for more information.


Copyright © 2026, Oracle and/or its affiliates.

### Service Point Consumption Extract Request Types and Requests

Configuration for service point consumption extracts are defined via Requests. See Understanding Requests in the *Application Framework Administrative User Guide* for more information about creating these records.

Service point consumption extract request types should be created using the SP Measurement Extract Request Type (D1-SPMsrmtExtractRequestType) business object, and define the following parameters:

- The Device Identifier Type (such as Asset Number, Badge Number, and so on) to include in the extract file
- File parameters, including:
  - **File Name**: String used as the prefix for extract file names
  - **File Location**: The location where extract files will be saved (see External File Storage in the *Application Framework Administrative User Guide*)
  - **Child Folder**: An optional child folder in the file location
  - **Record Limit**: A limit on the number of records to include in extract files

Service point consumption extract requests should be created using the SP Measurement Extract Request (D1-SPMsrmtExtractRequest) business object, and define the following parameters:

- **Start/End Date/Time**: The date and time range for the extract
- **Entity Group**: The entity to which the service points for which consumption is extracted are assigned
- **UOM/TOU/SQI**: The type of measurement data extracted, filtered based on the Service Type for the selected Entity Group
- **Device Identifier Type**: The device ID type to include in the extract file (such as Asset Number, Badge Number, and so on)
- **Consumption Type**: The type of consumption (Interval or Scalar) for extract
- **Target Interval Size**: The interval size for the extract, applicable only if Consumption Type is Interval. Note that the interval size must be larger than the SPI of the measurement data
- **Batch Number**: The batch number for the batch process used to perform the extract. This is populated by the batch process that performs the extract.

Extract files for Service Point Consumption Requests are created as external URL attachments on the Request that processed the extract.

### Interval Snapshot Mapping

Service point consumption extracts require interval snapshot mapping data be defined for the interval size and date range of the extract. See About Interval Snapshot Mapping and Maintaining Interval Snapshot Mapping for more information.

### Request Processing

Creating a request starts batch job to extract the usage to Oracle Object Storage or a file directory in a CSV format. This CSV format can easily be imported into Excel or other software.

When a request is created, it starts in the Pending state, and an Enter algorithm creates a deferred request and the SP Measurement Extract Request Monitor (D1-SPMRQ) batch


Copyright © 2026, Oracle and/or its affiliates.

process moves the request into the Extract state, where the Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm performs the extract process.

The extract process creates an "External URL" attachment in which the attachment contains a URL to the extract file in the location specified in the Request Type. Attachments created for a request can be accessed via the **Request Attachments** zone in the **Request** portal.

### File Naming and Formats (SECTION)

Extract file names use the following format:

```
<file location><file name>-<request id>-<start dttm>-<end dttm>_<file number>
```

where:

- **<file location>** is the **File Location** specified in the Request Type
- **<file name>** is the **File Name** specified in the Request Type
- **<request id>** is the ID of the Request
- **<start dttm>** is the start date and time for the request, displayed as YYYYMMDDHH24MISS
- **<end dttm>** is the end date and time for the request, displayed as YYYYMMDDHH24MISS
- **<file number>** is the file number for the extract file. This is a 4-digit number with padded leading zeros. Multiple files are created if a **Record Limit** is specified on the Request Type and the records extracted exceeds the limit.

For example if the **File Location** was set to "file-storage://CM-CloudStorage/sp-measurement-extracts/" and the **File Name** was set to "sp-kWh-extract" with an extract date/time of 2026-01-01 00:00:00 to 2026-02-01 00:00:00 then the first file would be named:

### Scalar Record Format

```
file-storage://CM-CloudStorage/sp-measurement-extracts/sp-kWh-
extract-10392739850273-20260101000000-20260102000000_0001.csv.gz
```

Scalar records will only have one measurement per row. The **Start Date/Time** isn't the start of the day but rather the start of the measurement. Measurements are extracted based on when they end, so the first measurement chronologically for each measuring component may cover a period that falls before the extract time period. Extract data comes from the following tables and columns:

| Header Field | Table | Column | Notes |
|---|---|---|---|
| Start Date/Time | D1_MSRMT | PREV_MSRMT_DTTM | Converted from the time zone defined in Installation Options to the service point time zone |
| Customer Name | D1_CONTACT_NAME | NAME_VALUE | This is for the "Main" usage subscription Contact and for the Primary name |
| Address | D1_SP | ADDRESS1 + ADDRESS2 + ADDRESS3 + ADDRESS4 | ADDRESS2 through 4 will be trimmed |


Copyright © 2026, Oracle and/or its affiliates.

| Header Field | Table | Column | Notes |
|---|---|---|---|
| Service Point | D1_SP | D1_SP_ID | |
| Local Time Zone | CI_TIME_ZONE | F1_TIMEZONE_NAME | The time zone from the service point |
| Measuring Component | D1_MEASR_COMP | MEASR_COMP_ID | |
| Meter Number | D1_DVC_IDENTIFIER | ID_VALUE | Based on the device identifier type to use (such as badge number, serial number, and so on) from the Request Type |
| How to Use Flag | D1_MEASR_COMP | MEASR_COMP_USAGE_FLG | |
| UOM | D1_MC_TYPE_VALUE | D1_UOM_CD | Unit of Measure (Description) |
| TOU | D1_MC_TYPE_VALUE | D1_TOU_CD | Time of Use (Description) |
| SQI | D1_MC_TYPE_VALUE | D1_SQI_CD | Service Quantity Identifier (Description) |
| Date/Time | D1_MSRMT | MSRMT_DTTM | Converted from the time zone defined in Installation Options to the service point time zone |
| Measurement | D1_MSRMT | MSRMT_VAL | The appropriate Measurement Value field based on the Measuring Component Type's Value Identifier that was selected using the UOM/TOU/SQI. This is the main measurement Measurement Value the majority of the time but for Gas or Water, derived values from the other measurement value fields may be requested. |
| Condition | D1_MSRMT | MSRMT_COND_FLG | |

### Interval Record Format

Interval records will have multiple measurements per row. Extract data comes from the following tables and columns:

| Header Field | Table | Column | Notes |
|---|---|---|---|
| Start Date/Time | D1_INT_SNAP_MAP | D1_LOCAL_DT | Converted from the time zone defined in Installation Options to the service point time zone |


Copyright © 2026, Oracle and/or its affiliates.

| Header Field | Table | Column | Notes |
|---|---|---|---|
| Customer Name | D1_CONTACT_NAME | NAME_VALUE | This is for the "Main" usage subscription Contact and for the Primary name |
| Address | D1_SP | ADDRESS1 + ADDRESS2 + ADDRESS3 + ADDRESS4 | ADDRESS2 through 4 will be trimmed |
| Service Point | D1_SP | D1_SP_ID | |
| Local Time Zone | CI_TIME_ZONE | F1_TIMEZONE_NAME | The time zone from the service point |
| Measuring Component | D1_MEASR_COMP | MEASR_COMP_ID | |
| Meter Number | D1_DVC_IDENTIFIER | ID_VALUE | Based on the device identifier type to use (such as badge number, serial number, and so on) from the Request Type |
| How to Use Flag | D1_MEASR_COMP | MEASR_COMP_USAGE_FLG | |
| UOM | D1_MC_TYPE_VALUE | D1_UOM_CD | Unit of Measure (Description) |
| TOU | D1_MC_TYPE_VALUE | D1_TOU_CD | Time of Use (Description) |
| SQI | D1_MC_TYPE_VALUE | D1_SQI_CD | Service Quantity Identifier (Description) |
| Total | D1_MSRMT | MSRMT_VAL<N> | The sum of the measurements for the Start Date/Time for the Measuring Component. The appropriate Measurement field will be used based on the Measuring Component Type's Value Identifier that was selected using the UOM/TOU/SQI. |
| Time<N> | D1_MSRMT | MSRMT_DTTM | This is date and time of each interval, calculated as follows: Start Date/Time (D1_LOCAL_DT) + (interval number * SPI). SPI is based on the **Target Interval Size** if it was provided in the Request or the **Interval Size** of the Measuring Component Type if not. |


Copyright © 2026, Oracle and/or its affiliates.

| Header Field | Table | Column | Notes |
|---|---|---|---|
| Measurement<N> | D1_MSRMT | MSRMT_VAL<N> | The appropriate Measurement Value field based on the Measuring Component Type's Value Identifier that was selected using the UOM/TOU/SQI. This is the main measurement Measurement Value the majority of the time but for Gas or Water, derived values from the other measurement value fields may be requested. |
| Condition<N> | D1_MSRMT | MSRMT_COND_FLG | This will either be the the Minimum (MIN) or Maximum (MAX) Measurement Condition Flag based on the configuration of the Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm. The default is to use the MIN.<br>Note: MIN or MAX is only important for Measuring Components that have a lower SPI than the **Target Interval Size**. If no **Target Interval Size** is provided then data is extracted based on the **Interval Size** of the of the Measuring Component Type. |

Interval data is values are returned as a single string of comma-separated values that are sub delimited by tildas (~). For example:

```
3.732~1~501000,5.804~2~501000,6.432~3~501000,5.572~4~501000,7.384~5~501000,7.3
58~6~501000,4.67772~7~501000,7.8132~8~501000,7.83564~9~501000,6.14856~10~50100
0,8.0988~11~501000,8.16816~12~501000,4.63186~13~501000,7.7366~14~501000,7.7588
2~15~501000,6.08828~16~501000,8.0194~17~501000,8.08808~18~501000,3.808~19~5010
00,4.448~20~501000,7.69~21~501000,6.94~22~501000,7.964~23~501000,7.316~24~5010
00
```

### Customizing Extract Output

The Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm uses the Extract Measurements for SP Entity Group Library (D1EMSPEGLib) script to return queries, format headers, and format records. This script implements three high level types of methods:


Copyright © 2026, Oracle and/or its affiliates.

1. Method to return the query that will extract the measurement data and master data details
    a. This takes in the same fields as are available in the plug-in driven script process data plug-in spot
    b. This returns a prepared statement
    c. There is one script for Interval and one script for Scalar since the queries are very different
2. Method to populate a header row for the CSV file
    a. For scalar this is always the same fields since we only ever extract one row at a time
    b. For interval this creates the measurement fields according to the maximum SPI or the target SPI and whether or not the extract period covers a DST transition
3. Method to process the returned data into the formatted row
    a. This takes in the results from executing the queries
    b. This formats the data into what will be streamed into the file and return that as a string

Use the **Script** portal to view this script in more detail.

To customize the data extracted via this process (for example to add an additional column), a custom (CM) version of the Groovy Library script can created. This involves the following steps:

1. Duplicate the Extract Measurements for SP Entity Group Library (D1EMSPEGLib) script.
2. Adjust the generate query method(s) to retrieve the additional information.
3. Adjust the generate header method(s) to return the new field for the header.
4. Adjust the result formatting method(s) to place the new data into the result string according to their requirements.
5. Add this new CM script as the value for the **Override Extract Groovy Library** parameter on the the Perform SP Measurement Extract (D1PMSPMSRMEX) algorithm.

# Consumption Extract Type


- Understanding Consumption Extract Type
- Configuring Consumption Extract Type


## Understanding Consumption Extract Type

The Consumption Extract Type controls which service point's measurements are extracted for Oracle Utilities DataConnect, what type of measurement is extracted, and how the measurements are grouped into TOU periods (if preferred).

The Consumption Extract Type also controls the request type used for creating Consumption Extract Requests of this type, how frequent automated requests are created for incremental extract, the batch jobs that are triggered for extracting data and the algorithms that extract, format, and write the data that is extracted.

Refer to the Oracle Utilities DataConnect section for more details on how this object is specifically put into use.

## Configuring Consumption Extract Type



Copyright © 2026, Oracle and/or its affiliates.

This portal is used to display and maintain a Consumption Extract Type.

Refer to Understanding Consumption Extract Types for more information.

You can access the portal by selecting **Admin**, then **Integration**, then **Consumption Extract Type**.

The following zones may appear as part of the portal's **Main** tab page:

- **Consumption Extract Type List**: displays all of the Consumption Extract Types so you can choose the one you want to display in more detail
- **Consumption Extract Type**: shows the specific configuration for the selected Consumption Extract Type

# Analytics Configuration

This portal provides a bird's eye view of the configuration information for Oracle Utilities Analytics. It provides links and guidelines for the areas that need configuration to successfully run the extract transfer and load processes from Oracle Utilities Analytics.

Refer to the *Oracle Utilities Analytics Administration Guide* for more information.

You can access the portal from the **Admin > Analytics Configuration > Analytics Configuration**.

The following zones may appear as part of the portal's **Main** tab page

- **Bucket Configuration List.** This zone lists bucket configurations to set up.
- **BI-Related Business Objects Information.** This zone lists additional control entities to set up.

The following zones may appear as part of the portal's **OWB-Based ETL** tab page

- **Outbound Sync BOs and Algorithm List.** This zone lists business objects and algorithms involved in the extract process.
- **BI-Oriented Extendable Lookup.** This zone lists extendable lookups to set.
- **External Data Source Indicators List.** This zone lists the external identifiers of the various sources for your analytics data.

**Note:** Service Point business objects can make use of the following System Events:

- **Service Point Snapshot**: This system event defines the algorithm used to create a snapshot of the Service Point for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - Snapshot."
- **Usage Snapshot**: This system event defines the algorithm used to create a usage snapshot of the Service Point for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - Usage Snapshot."
- **Unreported Usage Analysis Snapshot**: This system event defines the algorithm used to create a snapshot of the Service Point's consumption since the last usage transaction for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - Unreported Usage Analysis Snapshot."
- **SP VEE Exception Snapshot**: This system event defines the algorithm used to create a snapshot of VEE exceptions for the Service Point for use with Oracle Utilities Analytics. The Algorithm Entity for available algorithms is "SP (BO) - VEE Exception Snapshot."


Copyright © 2026, Oracle and/or its affiliates.

# Migrating Data Extract Processing from Legacy Measurement Processing to Direct Measurement Processing

When processing data extracts using Direct Measurement Processing, change data capture information is recorded in the Measurement Extract Request table instead of the General Process table. This requires that the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration be set to "Yes". This applies to all types of data extracts, including:

- Configurable Consumption Extracts
- Measurement Data Extracts
- Initial Measurement Data Header Extracts
- Aggregated Measurement Data Extracts
- DataConnect Consumption Extracts (see Oracle Utilities Dataconnect / Opower for more information about DataConnect extracts)

For customers who had been using Legacy Measurement Processing and processing data extracts (in which change data capture information is recorded in the Genera Process table), migrating to using Direct Measurement Processing will involve transitioning out of Legacy-based extract processing and into Direct Measurement Processing extract processing.

There are two approaches to this transition, one involves continuing to use the current batch controls, the other involves creating new batch controls.

### Using Existing Batch Controls

To use existing batch controls, do the following:

1. Pause all Initial Measurement Data Processing.
2. Run each of the Historical extract processes to process all remaining records in the General Process table.
3. Enable use of the Measurement Extract Request table by setting the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration be set to "Yes".
4. Flush the batch server.
5. Resume Initial Measurement Data processing.

### Creating New Batch Controls

To create new batch controls, do the following:

1. Duplicate all of the Historical extract batch controls
2. Update the Initial Measurement Data algorithms and Consumption Extract Types to reference the new batch controls.
3. Flush the batch server.
4. Process one final run using the original Historical extract batch controls.
5. Enable use of the Measurement Extract Request table by setting the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration be set to "Yes".


Copyright © 2026, Oracle and/or its affiliates.

6. Start using the new batch controls for all extract processing.

# Information Lifecycle Management (ILM)


- Understanding Information Lifecycle Management (ILM)
- Configuring Information Lifecycle Management (ILM)


## Understanding Information Lifecycle Management (ILM)

Information Lifecycle Management (ILM) is a separately licensable component of Oracle Utilities Meter Data Management that works in tandem with Oracle database feature of the same name.

For further background on the overall ILM process see the Information Lifecycle Management chapter of the Oracle Utilities Application Framework *Administrative User Guide*.

This section provides an overview of the components used by the Information Lifecycle Management functionality, including:

- ILM-Enabled Maintenance Objects: Single Retention Period
- ILM-Enabled Maintenance Objects: Multiple Retention Periods
- Adjusting Eligibility of Non-Final Transactions
- Archive Eligibility Hierarchy
- Multiple Retention Period Eligibility Crawling Strategies

Refer to the Information Lifecycle Management chapter of the Oracle Utilities Application Framework *Administrative User Guide* for general information about how ILM works with OUAF applications.

**ILM-Enabled Maintenance Objects: Single Retention Periods**

The majority of Maintenance Objects support a single retention period. They can either inherit the system wide retention days that are configured on the ILM Configuration master configuration or they can have a retention period specified via the Maintenance Object option ILM Retention Period in Days.

For these objects either the Maintenance Object specific retention period or the system wide retention period will apply to all transactions.

**ILM-Enabled Maintenance Objects: Multiple Retention Periods**

For several Maintenance Objects the retention period may vary based on the type of transaction. For example, initial measurement data for non-billed units of measure such as voltage might be archived much more rapidly than measurements that feed into the production of billing determinants.

These maintenance objects are differentiated from the single retention period maintenance objects in a few ways:

- The primary table for the entity has an additional column Retention Period (RETENTION_PERIOD) that captures the retention period in days for each transaction. This value is populated based on the ILM Configuration - MDM master configuration.
- There is additional configuration available on the ILM Configuration -MDM master configuration to define multiple retention periods by maintenance object specific criteria



Copyright © 2026, Oracle and/or its affiliates.

- Initial Measurement Data: retention periods can be specified by type of IMD (Interval vs Scalar), and primary UOM of the measuring component
- Device Events: retention periods can be specified by device event type category
- Activities: retention periods can be specified by activity type category
- There is a special ILM Crawler batch that supports crawling transactions by their specific retention periods
- A combination of each transactions ILM Date and Retention Period are considered when identifying records to evaluate for eligibility. Refer to batch controls ILM Crawler - Device Event (D1-DECRL), ILM Crawler - IMD (D1-IMDCL), or ILM Crawler - Activity (D1-ACTCR) for more details.
- There is an additional maintenance object option ILM Date Partition Months which feeds into the ILM Crawler batch to delay processing until all records on a partition would be eligible. This is explained later in this section under the header Multiple Retention Period Eligibility Crawling Strategies.

### Adjusting Eligibility of Non-Final Transactions

As delivered Oracle Utilities Meter Data Management and Customer to Meter maintenance objects allow non-final transactions to be eligible for archive. This is a logical approach when transactional data has a very long retention period (e.g. 3 year old non-final Initial Measurement Data is unlikely to have relevance and in many circumstances shouldn't be finalized).

If non-final transactions should not be eligible for archive this can be controlled at either the Maintenance Object (MO) or Business Object (BO) level:

- **MO - ILM Restrict By Status (Y/N)**: Setting this to "Y" opts into the ability to restrict archiving by status, this is required to be set for either of the MO or BO level restrictions to be considered. To enforce the status it requires that either the **ILM Restrict By BO Final Status (Y/N)** or **ILM Final Status Field Value** options be configured as well.
- **MO - ILM Restrict By BO Final Status (Y/N)**: Setting this to "Y" indicates the transactions for the MO must be in a final status to be eligible for ILM. A status is considered to be final based on the BO lifecycle configuration. This can be overridden by the BO option **Final Status Required for Archive (Y/N)**. This is only applicable for an MO that is maintained by business objects.
- **BO - Final Status Required for Archive (Y/N):** Setting this to "Y" indicates the transactions for this particular BO must be in a final status. This setting overrides the MO level **ILM Restrict By BO Final Status (Y/N).**

The following table helps to illustrate how these three settings impact whether a given transaction must be in a final status:

| MO: ILM Restrict By Status | MO: ILM Restrict By BO Final Status | BO: Final Status Required for Archive | Final Status Required for Eligibility? |
|---|---|---|---|
| N | Y | Y | N |
| N | Y | N | N |
| N | N | Y | N |
| Y | N | N | N |
| Y | Y | N | N |
| Y | Y | Y | Y |
| Y | N | Y | Y |


Copyright © 2026, Oracle and/or its affiliates.

| MO: ILM Restrict By Status | MO: ILM Restrict By BO Final Status | BO: Final Status Required for Archive | Final Status Required for Eligibility? |
|---|---|---|---|
| Y | Y | Not Provided | Y |
| Y | N | Not Provided | N |

> **Note**
>
> There are a few MOs that do not support the BO level setting. To best understand how these fields are interpreted for an MO, refer to the detailed description of the algorithm plugged into the ILM Eligibility system event for the MO. For example, the generic ILM Eligibility Based on Status (F1-ILMELIG) algorithm does not support the BO level override option.

### Archive Eligibility Hierarchy

The table below summarizes details of the eligibility algorithms for each maintenance object supported by ILM. Information on this table includes:

- Cascaded By: Indicates how archiving for transactions for the maintenance object is initiated. For example, device events can be archived by themselves or as part of a related activity.
- ILM Date Computation: Indicates the type(s) of transactions that impact the calculation of the ILM Date for transactions for the maintenance object. "Itself" indicates that the ILM date is not impacted by other transactions.
- Specific Eligibility Considerations: Indicates specific considerations used by the eligibility algorithm.
- Cascaded Transactions: Indicates other transactions that are archived when transactions for the maintenance object are archived. For example, VEE exceptions are archived with initial measurements.

| Maintenance Object | Cascaded By | ILM Date Computation | Specific Eligibility Considerations | Cascaded Transactions |
|---|---|---|---|---|
| Initial Measurement | | Create date | VEE Exceptions must be eligible<br>Associated Related MC Synchronization activity must be complete | VEE Exceptions |
| VEE Exception | Initial Measurement | Earlier of related Initial Measurement or VEE Exception Create Date | Related Service Issue Monitor must be complete | |
| Usage Transaction | | The latest create date for all related usage transactions | All related usage transactions must be eligible for archiving | Related Usage Transactions |


Copyright © 2026, Oracle and/or its affiliates.

| Maintenance Object | Cascaded By | ILM Date Computation | Specific Eligibility Considerations | Cascaded Transactions |
|---|---|---|---|---|
| Device Event | Activity | Create date<br>Superseded by paired event create date if earlier (where applicable) | Related paired events must be eligible<br>Related service issue monitor must be complete<br>Related activity must be complete | |
| Activity | Parent activity | Create date<br>Superseded Device event, child activity, or completion event create date supersede (where applicable) | Cascaded transactions must be eligible<br>Command requests must not be associated to a non-final service issue monitor<br>Command requests must not be associated to an install event on off history entry | Child activities<br>Device events<br>Outbound communications<br>Inbound communications<br>Completion events |
| Outbound Communication | Activity | Create Date<br>Superseded by initiating activity's create date (where applicable) | | Inbound communication |
| Inbound Communication | Outbound Communication | Create date<br>Superseded by initiating outbound communication create date (where applicable) | | Device events |

### Proactive Archive Eligibility for Transactional Data

Some types of transaction data, including Device Events, Initial Measurement Data, and Usage Transactions can be analyzed for archive eligibility as they are received rather than waiting until their retention period has expired. This means that these records are judged ready to be archived as they are created and do not need to be read, analyzed, and updated during the ILM crawling process. This greatly reduces the number of records to be crawled at the end of the retention period and shortens the window of time required for ILM crawling. This section outlines how the ILM Archive switch (ILM_ARCH_SW) is set to “Y” (Yes) for records of these types as part of their initial processing.

- **Device Events**:
  - The ILM Archive switch is set to “Y” for device events that meet the following criteria:
    - No service issue monitors were created for the device event
    - The device event is not a paired device event


Copyright © 2026, Oracle and/or its affiliates.

- The ILM Archive switch is set to “Y” for device events via the following algorithms:
  - Create Service Issue Monitor from Device Event (D1-DVCEVTSIM)
  - Create Service Issue Monitor from Device Event for Paired Event (D1-PRDVEVSIM)

- **Initial Measurement Data**:
  - The ILM Archive switch is set to “Y” by default for initial measurements (this is a different approach than the other record types).
  - The ILM Archive switch is set to “N” for initial measurements that meet the following criteria:
    - A consumption synchronization activity has been created for the initial measurement
    - A service issue monitor initiated from a VEE exception related to the initial measurement
  - The ILM Archive switch is set to “N” for initial measurements that meet the above criteria via the following algorithms:
    - Update Latest Measurement Date/Time on Scalar MC with Consumption Sync (D1-UDTSCMCRE)
    - Update Latest Measurement Date/Time on MC with Consumption Sync (D1-UPD-DTMC)
    - Create Service Issue Monitor from VEE Exception (D2-VEEEXCSIM)

- **Usage Transactions**:
  - The ILM Archive switch is set to “Y” when a usage transaction enters the “Sent” or “Discarded” state (or the “Calculated” or “Discarded” states for a sub-usage transaction ) via the following algorithms:
    - Send Usage (D2–SEND-USG)
    - Set ILM Switch to ‘Y’ (D1-SETILMSWY)

#### Multiple Retention Period Eligibility Crawling Strategies

When multiple retention periods are defined for a maintenance object each record will have its retention period written to the database at the time of record creation. This retention period is then used to identify when the record will be ready for eligibility evaluation (aka crawling).

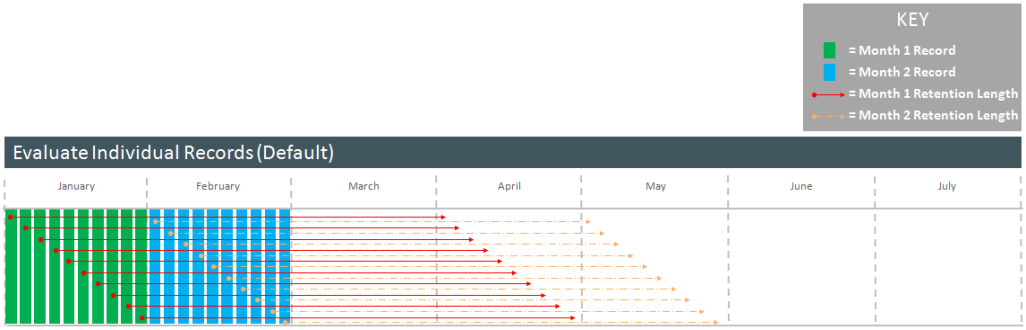
> **Note**
>
> As specified in the Oracle Utilities Meter Data Management Database Administrator's Guide each unique retention period for the maintenance object will be a sub-partition within a given ILM date partition.

The system supports two methodologies for doing so:

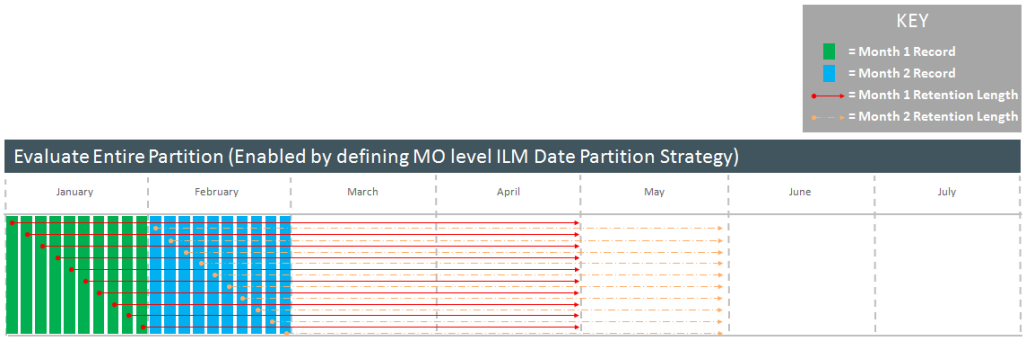
- Evaluate Individual Records:
  - This is the default approach.
  - This is the standard approach to evaluating ILM retention for all maintenance objects with a single retention period.


Copyright © 2026, Oracle and/or its affiliates.

- The ILM date for each record will be compared against the current date less the retention period for that type of record. Simply stated, any records older than the retention period will be evaluated for eligibility.
- In this option, a portion of a partition is processed each day. It is only after the entire partition is processed that the DBA can archive the partition
- This results in many visits to the partition, to determine eligibility.
- The image below illustrates how the 'evaluate individual records' method will crawl one day of records on a partition and sub-partition each day until the retention period for the last day of the partition has expired. All records would then be archived together when the last day of the partition is marked as eligible. This results in daily processing of small amounts of records but frequent visits to the partition.




- Evaluate the Entire Partition:
  - This option is enabled by configuring the ILM Date Partition Months maintenance object option.
  - In this option, all records on an ILM date partition retention period sub-partition will be evaluated at the same time.
  - For example, for a monthly partitioning strategy in the month of January, records in the January partition will not be evaluated until the January 31 records have reached the end of their retention period.
  - This means that all records for the 1st through the 31st are evaluated at one time.
  - This option typically results in a single visit to the partition, to determine eligibility
  - The image below illustrates how the 'evaluate entire partition' method will only crawl records from a given partition and sub-partition when all records on that partition are eligible to be evaluated. Therefore, when the last day of the partition is ready to be evaluated, the entire partition will be evaluated. This results in periodic processing with, and in many cases, a single visit to the partition for ILM purposes.





Copyright © 2026, Oracle and/or its affiliates.

# Configuring Information Lifecycle Management (ILM)

For additional information on configuring ILM, see Enabling ILM for Supported Maintenance Objects in the Oracle Utilities Application Framework *Administrative User Guide*.

The tasks described in the OUAF guide must be applied in conjunction with the task of configuring the Oracle Utilities Meter Data Management specific master configuration.

You can access the portal from by selecting **Admin**, then **General**, then **Master Configuration**.

Once the Master Configuration screen returns, locate the following record:

- **ILM Configuration - MDM:** this enables multiple retention periods to be configured for Initial Measurements, Device Events, and Activities.

### Related Eligibility Algorithms

There are a several Oracle Utilities Meter Data Management specific algorithms involved with ILM refer to the detailed description of the algorithm type of each algorithm for more information about the functionality each provides:

1. **ILM Eligibility - Activity** (D1-ILMELGACT): determines eligibility of Activities and cascades child Activities, Communication Out, Communication In, Device Events, and Completion Events.
2. **ILM Eligibility - Communication In** (D1-ILMELGCI): determines eligibility of Communication In.
3. **ILM Eligibility - Communication Out** (D1-ILMELGCO): determines eligibility of Communication Out and cascades Communication In.
4. **ILM Eligibility - IMD** (D1-ILMELGIMD): determines eligibility of Initial Measurement data and cascades VEE Exceptions.

> **Note**
>
> If applicable, update the **Populate measurement reading with IMD read when archiving (true/false)** parameter on this algorithm to `true` before running the ILM Crawler - IMD (D1-IMDCL) batch control.
> Setting this parameter to "true" instructs the algorithm to populate the measurement reading value from the Initial Measurement Data (IMD) while D1-IMDCL processes the records. This configuration is required only when the measurement reading value exists in the Initial Measurement Data (IMD) but is not populated in the corresponding measurement record. Customers should confirm this condition before changing the D1-ILMELGIMD parameter to `true`. This setting is not required when the measurement records already contain the reading value.

> **Note**
>
> If an Initial Measurement Data record is marked as ready for archive and a corresponding control table record still exists this algorithm will delete the control table record.


Copyright © 2026, Oracle and/or its affiliates.

5. **ILM Eligibility - Device Event** (D1-ILMELGDE): determines eligibility of Device Event data and cascades VEE Exceptions.
6. **ILM Eligibility - Usage Transaction** (D1-ILMELIGUT): determines eligibility of Usage Transactions.

> **Note**
>
> If a Usage Transaction record is marked as ready for archive and a corresponding control table record still exists this algorithm will delete the control table record.

7. **ILM Eligibility - Usage Transaction Exception** (D1-ILMELGUEX): determines eligibility of Usage Transaction Exceptions.
8. **ILM Eligibility - VEE Exception** (D1-ILMELGVEX): determines eligibility of VEE Exceptions.

The following provides additional information beyond that provided in the ILM Eligibility - Activity (D1-ILMELGACT) algorithms' detailed descriptions on how specific activity types are handled:

- Request Orchestration Activities: Request orchestrations also archive any child activities that were created, and those child activities archive any child activities or data (completion events, device events, etc.).
- Field Activities / Command Activities: Field activities and command request activities are archived either as part of a request orchestration or by themselves. When a field activity/ command request is archived it also archives any of the following child transactions:
  - Update/Cancel Orchestrators
  - Communication Out
  - Communication In
  - Completion Event
- Non-Dispatchable Activities: Non-Dispatchable activities archived either as part of a request orchestration or by themselves. When a non-dispatchable activity is archived it also archives child completion events.
- Orchestration Maintenance Activities: Orchestration Maintenance activities are only archived by themselves if they do not have a related activity. When an orchestration maintenance activity is archived it also archives any of the following child transactions:
  - Communication Out
  - Communication In
- Device Event Activities: Presently device event activities are limited to outages which have an initiating and an ending device event. These types of activities cannot be archived until their related device events are either archived or ready to archive. However, the device events do not have to wait for the activity to archive so the device events are not updated with the activity's ILM date.
- Bulk Activities: Bulk Activities are comprised of the Bulk Header and the Bulk Request/ Response, and depending on how the header was created there will be one header to many bulk requests/responses. The Bulk Header will only be eligible for archive if all related Bulk Request activities are also eligible for archiving. All related bulk activities are archived together. Bulk activities also generate one-to-many command request activities. Those individual command activities are archived separately.


Copyright © 2026, Oracle and/or its affiliates.

- Extract Request Activities: These activities are used to request data from the head-end system on a periodic basis. Extract request activities should be in a final state prior to being archived.
- Other Activities: Other types of activities can be archived provided they do not require any special logic for handling for archiving purposes, such as checking for related data. These activity types can archive without checking related transactions:
  - Consumption Sync
  - Dimension Scanner
  - Error Activity
  - Measurement Quantity (deprecated)
  - Meter Read Download Activity
  - Suppression
  - Usage Transaction Correction Processor

In the event that your implementation uses custom activity types that require special handling, a custom algorithm should be created and added prior to the base package algorithm to preemptively handle the activity type category.

### Related Batch Controls

There are a several Oracle Utilities Meter Data Management specific batch controls involved with ILM:

1. **ILM Crawler - Activity** (D1-ACTCR): identifies and executes eligibility evaluation for Activities. This batch control supports multiple retention periods.
2. **ILM Crawler - Communication In** (D1-CICRL): identifies and executes eligibility evaluation for Communication In.
3. **ILM Crawler - Communication Out** (D1-COCRL): identifies and executes eligibility evaluation for Communication Out.
4. **ILM Crawler - Device Event** (D1-DECRL): identifies and executes eligibility evaluation for Device Events.
5. **ILM Crawler - IMD** (D1-IMDCL): identifies and executes eligibility evaluation for Initial Measurement Data. This batch control supports multiple retention periods.
6. **ILM Crawler - Usage Transaction Exception** (D1-UEXCL): identifies and executes eligibility evaluation for Usage Transaction Exceptions.
7. **ILM Crawler - Usage Transaction** (D1-UTCRL): identifies and executes eligibility evaluation for Usage Transactions.
8. **ILM Crawler - VEE Exception** (D1-VEXCL): identifies and executes eligibility evaluation for VEE Exceptions.

For additional details see the Batch Processes section the Oracle Utilities Application Framework *Administrative User Guide*.

# Measurement Reprocessing

- Configuring Measurement Reprocessing


Copyright © 2026, Oracle and/or its affiliates.

# Configuring Measurement Reprocessing

This section describes the process for configuring Measurement Reprocessing.

Refer to the About Measurement Reprocessing section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

### Activity Type Configuration

Configuration for Measurement Reprocessing is performed by adding or changing an Activity Type. You can access the portal by selecting **Admin**, **Communication**, **Activity Type**.

Once the Activity Type search screen returns, locate the following records:

- **Measurement Reprocess Activity - Interval:** this Activity Type handles measurement reprocessing for interval Measuring Components.
- **Measurement Reprocess Activity - Scalar:** this Activity Type handles measurement reprocessing for scalar Measuring Components.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field. By adding these Activity Types you are activating the process within Oracle Utilities Meter Data Management that will monitor changes to either the Measuring Component multiplier (the **Register Multiplier** for register measuring components or **Channel Multiplier** for interval and auto-read register measuring components) or the Install Event's **Installation Constant** or **Installation Date/Time**. If any of these attributes change for a device, then a new Activity will be created that attempts to reprocess the measurements for the affected period.

### Algorithm Configuration

The Detect Installation Constant Change for Install Event (D1-DETIECHG) algorithm detects changes to install events, including changes to the **Installation Constant**, changes to the **Installation Date/Time**, and creation of new install events whose date range includes preexisting measurements and which include a change to the **Installation Constant**. When the algorithm detects a change to any of these, it initiates measurement reprocessing.

The "Ignore Install Event Date Changes" parameter can disable automatic initiation of measurement reprocessing in the event of changes to the **Installation Date/Time**.

The "Ignore New Install Events" parameter can disable automatic initiation of measurement reprocessing in the event of a new install event being created with a date range that includes preexisting measurements with a different **Installation Constant**.

### Related Batch Controls

There are a few batches involved with Measurement Reprocessing:

- **Measurement Reprocessing Monitor** (D1-MRAC): this processes any new Measurement Reprocessing activities that are created.
- **Measurement Reprocessing - Retry Monitor** (D1-MRER): this process retries any Measurement Reprocessing activities that fail to the Issue Detected state.

# Outage Storm Mode


Copyright © 2026, Oracle and/or its affiliates.

- Configuring Outage Storm Mode
- Detailed Examples of Outage Storm Mode


# Configuring Outage Storm Mode

This section describes the process for configuring Outage Storm Mode.

> **Note**
>
> Refer to the About Outage Storm Mode section of the *Oracle Utilities Meter Solution Business User Guide* for more functional information.

### Master Configuration settings

Configuration for Outage Storm Mode is performed by adding or changing a Master Configuration. You can access the portal from the **Admin > General > Master Configuration**.

Once the Master Configuration search screen returns, locate the **MDM Master Configuration**. Use the Add button beside the record to configure for the first time. If a record has already been added, then click the Edit button instead. The **Estimation Eligibility For Widespread Outages** section is the primary section that needs to be configured for Outage Storm Mode to be enabled. Use the embedded help to guide you through the meaning of each configuration field.

### Setup Meter Communication Tracking Aggregation

As part of Outage Storm Mode, aggregations are performed to determine the percentage of meter communication that has occurred for an area. The following configuration supports this:

1. Access the portal for Measuring Component Type through **Admin > Device > Measuring Component Type**. Add a new Measuring Component Type using the **Meter Communication Tracking Aggregator Type** (D2-MtrCommTrckngAggregatorType) business object. Follow the embedded help provided for each section. Two areas listed below are crucial to preparing for this specific aggregation:
    a. When selecting a Measuring Component Business Object for this record, choose the **Meter Communication Tracking Aggregator** Business Object.
    b. For the Value Identifiers, there are a few metrics that are can be defined as listed on the D2-AGG-MCTM Algorithm Type:
        i. Read Percentage
        ii. Count of Received Measuring Components
        iii. Count of Total Measuring Components
2. Access the portal for Activity Type through Admin > Communication > Activity Type. Add a new record for the Meter Communication Tracking Dimension Scanner Activity Type.
3. Access the portal for Activity through **Main > Communication > Activity**. Add a new record for the Activity Type you defined in the prior step.

### Standard Event Name Configuration

If there is a desire to indicate that a widespread outage has ended for a device when it receives certain device events, the **Standard Event Name** Extendable Lookup can be used for this. You can access the portal from the **Admin > General > Extendable Lookup**.



Copyright © 2026, Oracle and/or its affiliates.

Any Device Event that should close out Estimation Suppression Activities must be configured with the "End Estimation Suppression" algorithm (D2-EN-ESTSUP) in the Additional Processing Algorithms section. This is often configured for "Power Up" events to indicate that normal power flow has resumed for the meter.

### Related Batch Controls

There are a few batches directly involved with Outage Storm Mode:

- **Dimension Scanner** (D1-ADS): monitors the "Meter Communication Tracking Dimension Scanner" business object. New Aggregator Measuring Components will be created for every applicable Service Type, Postal Code, and Head End in the system.
- **Aggregation Monitor** (D2-AGG): runs to execute the logic for the "Meter Communication Tracking Aggregator" Measuring Component. This logic will aggregate all meters for the defined Service Type, Postal Code, and Head End. Once the read percentage is found for that day then a measurement will be created for the Aggregator Measuring Component.
- **Estimation Suppression Monitor** (D1-ESTSU): This batch monitors the Estimation Suppression activities for processing. This batch should be scheduled to run regularly so it can clean up Suppression Activities if they have ended.
- **Widespread Outage** (D2-WSO): This batch queries the Meter Communication Tracking Aggregators to determine if any have experienced a widespread outage. For any actively in outage mode, then the batch will find all related Service Points and execute the logic on the algorithm plugged into the BO system event of "Estimation Suppression". It's recommended that D2-WSO be scheduled shortly after aggregation D2-AGG.

## Detailed Examples of Outage Storm Mode

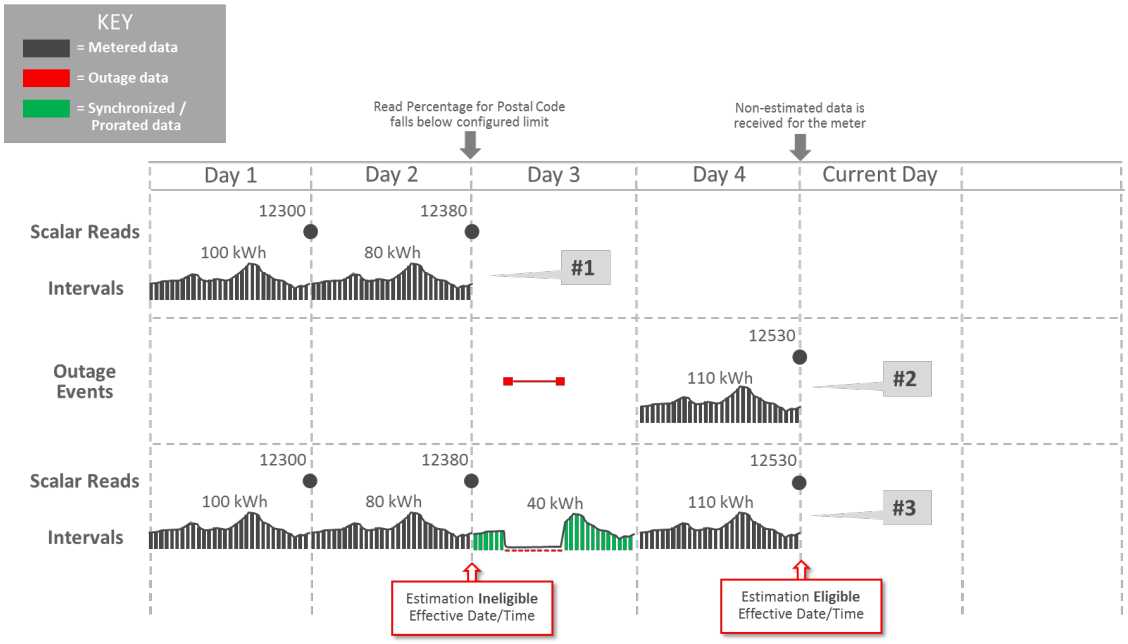
### Scenario 1: Meter is marked estimation ineligible, then metered data is received with outage events

The following configuration settings should be assumed for this example:

| Master Config option | Value |
|---|---|
| Check Widespread Outages | Yes |
| Read Percentage Days to Examine | 7 |
| Percent Drop From Norm - Estimation Ineligible | 30.0% |
| Minimum Device Count For Ineligibility | 100 |
| Days Before Becoming Ineligible | 0 |
| Fill Missing Data With Zero | No |
| Days Before Filling Zero | |
| Maximum Estimation Ineligibility Days | 3 |
| Percent Within Norm - Resume Estimation | 5.0% |

Diagram:


Copyright © 2026, Oracle and/or its affiliates.




Explanation for diagram:

1. The meter was communicating consistently but then a widespread outage occurs in the same postal code so the meter has an estimation suppression created.
2. Two events are received...a Power Down event and a Power Up event. Also, Regular usage data is received from the meter. The reception of non-estimated data marks the meter as eligible for estimation again.
3. Oracle Utilities Meter Data Management estimates data and takes the two events into account. Intervals marked as outage and filled with zeros are created during the outage period by the Device Periodic Estimation (D1-DVCPE) batch.

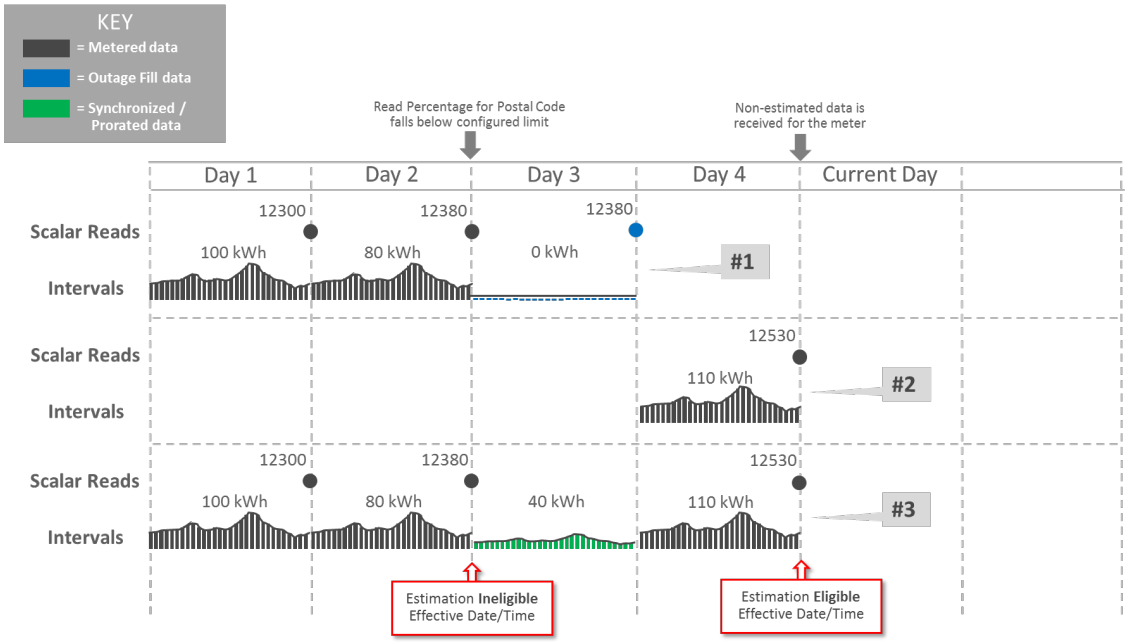
### Scenario 2: Meter is marked estimation ineligible, is filled with zeros, then receives actual data

The following configuration settings should be assumed for this example:

| Master Config option | Value |
|---|---|
| Check Widespread Outages | Yes |
| Read Percentage Days to Examine | 7 |
| Percent Drop From Norm - Estimation Ineligible | 30.0% |
| Minimum Device Count For Ineligibility | 100 |
| Days Before Becoming Ineligible | 0 |
| **Fill Missing Data With Zero** | **Yes** |
| **Days Before Filling Zero** | **0** |
| Maximum Estimation Ineligibility Days | 3 |
| Percent Within Norm - Resume Estimation | 5.0% |

Diagram:


Copyright © 2026, Oracle and/or its affiliates.




Explanation for diagram:

1. The meter loses communication and periodic estimation fills in outage intervals.
2. The meter reestablishes communication and sends in the latest readings and interval data.
3. Oracle Utilities Meter Data Management creates new data that's synchronized to fall in line with the new data from the meter by the Consumption Sync batch. This data replaces the outage measurements as it balances out to the scalar and interval data.

# Running Initial Periodic Estimation

This section provides guidelines for running an initial periodic estimation process.

- Overview
- Avoiding Long Execution Time
- Overview of Options and Start of Estimation Horizon
- Testing Periodic Estimation on Small Data Sets
- Testing Periodic Estimation on Large Data Sets

## Overview

Prior to running the first execution of periodic estimation an assessment should be made regarding the current state of measurement data on your measuring components. Without proper preparation the first execution can take an extremely long time due to large gaps on measuring components. Since the Device Periodic Estimation (D1-DVCPE) batch process is threaded by device, all of the initial measurements for a device's interval and scalar channels will be created within a single work unit. This will lead to an abnormally large number of transactions being created within a single commit which can degrade performance.

There are several options available for establishing a consistent starting point for estimation across all measuring component types:


Copyright © 2026, Oracle and/or its affiliates.

1. Configure the **Maximum Days to Estimate** on each periodic estimation eligible measuring component type. This limits the number of days periodic estimation will look backwards to create estimations. For example, if set to 60 the no estimations over 60 days old would be created and all measuring components would have their **Last Contiguous Measurement Date Time** set to 60 days ago or more recent.
2. Configure the **No of Hours in the Past to Retrieve Last Usable Measurement** option on the Measurement Data Options feature configuration. This configuration works in two specific ways:
   - When the measuring component's **Last Contiguous Measurement Date/Time** has not been initialized then this setting is used to search for usable measurements starting from the **Process Date/Time** less the number of hours configured.
     - If a measurement is found, the most recent measurement within that window will be used.
     - If no measurements are found, estimates will be generated starting from the calculated start date/time.

     > **Note**
     >
     > Note: if the calculated start date/time is very recent and not yet eligible for estimation then the measuring component's **Latest Contiguous Measurement Date/Time** will be updated with that value.

   - When the **Last Contiguous Measurement Date/Time** has already been set for a measuring component this setting will be used similar to the **Maximum Days To Estimate** and will limit how far back estimation will occur.
3. Use the **Initialize Measuring Component Dates** (D1-IMCDT) batch control just prior to the first run of periodic estimation to explicitly set the **Latest Contiguous Measurement Date/Time** across all measuring components to the date you would like to estimate from.
4. The **Perform Changes in New Session** parameter on periodic estimation algorithms (including D1-CIITBIH and D1-ARSPE) can improve processing when many initial measurements are being created in one run of these algorithms. When this parameter is set to "Y" each initial measurement will be created in a separate session and will be committed as they are completed. This improves performance since there are fewer changes being cached and ensures progress is saved in the event of errors during processing.

It is important to take into consideration that the periodic estimation process is tuned with the expectation that it will create estimations for a small percentage of the meter population (generally around 2%) for one day. If on the initial run all devices have measuring components that require estimations to be created, it is increasing the typical work load exponentially: 100% of the meter populate for many days.

Lastly, it is very important, regardless of the method used, to test a small population of meters before running the Device Periodic Estimation (D1-DVCPE) batch process on your entire meter population. Isolated testing will alert you to potential performance issues without having to cancel a multi-day D1-DVCPE execution.

## Avoiding Long Execution Time

In general the single most important thing to avoid long execution times on your first run of the Device Periodic Estimation batch process (D1-DVCPE) is to ensure that your measurement data is as current as it possibly can be. You should not run the Device Periodic Estimation


Copyright © 2026, Oracle and/or its affiliates.

batch process (D1-DVCPE) until you have completed conversion and you have loaded your initial set of initial measurements. If you run the Device Periodic Estimation batch process (D1-DVCPE) any time prior to that you risk generating estimates for data that you have but just haven't loaded yet.

If you must run the Device Periodic Estimation batch process (D1-DVCPE) prior to having all of your measurement data current, you should use the **Batch Business Date** on the batch control to limit the horizon of time you will estimating. The **Batch Business Date** won't impact the start of your horizon but it will limit the end.

The following sections provide input on several ways you can control the start of the horizon.

## Overview of Options and Start of Estimation Horizon

The table below summaries various options and their impact on the Start of the Estimation horizon.

> **Note**
>
> Always ensure the **Most Recent Measurement Date/Time** is set to the measurement date/time furthest into the future. The **Initialize Measuring Component Dates** (D1-IMCDT) batch control has functionality for initializing the Most Recent Measurement Date/Time as well (if it was not set during conversion).

| Measuring Component Type Maximum Days to Estimate | Feature Configuration | Measuring Component Dates | Measurement Data | Start of Estimation Horizon |
|---|---|---|---|---|
| Not Configured | Not Configured | Not Set | None | Horizon will begin based on the installation date/time:<br>• Interval uses the installation date/time<br>• Scalar uses the On event |
| Not Configured | Not Configured | Not Set | Yes, with some gaps | Horizon will begin based on the installation date/time:<br>• Interval uses the installation date/time<br>• Scalar uses the On event |
| Configured | Any | Not Set | Any | Horizon will begin based on the process date/time less the **Maximum Days to Estimate** |
| Configured | Any | Set | Yes - prior to Maximum Days to Estimate | Horizon will begin based on the process date/time less the **Maximum Days to Estimate** |


Copyright © 2026, Oracle and/or its affiliates.

| Measuring Component Type Maximum Days to Estimate | Feature Configuration | Measuring Component Dates | Measurement Data | Start of Estimation Horizon |
|---|---|---|---|---|
| Configured | Any | Set | Yes - after Maximum Days to Estimate | Horizon will begin based on the **Latest Contiguous Measurement Date/ Time** |
| Not Configured | Configured | Not Set | None | Horizon will begin based on the process date/time less the hours configured on the feature configuration.<br>If it is not yet time to estimate based on the calculated start of the estimation horizon, then:<br>• Interval: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start<br>• Scalar: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start and **Adjusted Last Contiguous Measurement Date/Time** is set to the appropriate value based on the periodic estimation configuration.<br>Note: this would happen when the **Hours Before Estimation** (for example 72) is a larger number of hours than the **No of Hours in the Past to Retrieve Last Usable Measurement** (for example 24) |


Copyright © 2026, Oracle and/or its affiliates.

| Measuring Component Type Maximum Days to Estimate | Feature Configuration | Measuring Component Dates | Measurement Data | Start of Estimation Horizon |
|---|---|---|---|---|
| Not Configured | Configured | Not Set | Yes - outside feature configuration window | Horizon will begin based on the process date/time less the hours configured on the feature configuration.<br>If it is not yet time to estimate based on the calculated start of the estimation horizon, then:<br>• Interval: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start<br>• Scalar: **Last Contiguous Measurement Date/Time** is initialized to the calculated horizon start and **Adjusted Last Contiguous Measurement Date/Time** is set to the appropriate value based on the periodic estimation configuration.<br>Note: this would happen when the **Hours Before Estimation** (for example 72) is a larger number of hours than the **No of Hours in the Past to Retrieve Last Usable Measurement** (for example 24) |
| Not Configured | Configured | Not Set | Yes - inside feature configuration window | Horizon will begin as of the most recent measurement date/time that was found for the measuring component. |


Copyright © 2026, Oracle and/or its affiliates.

| Measuring Component Type Maximum Days to Estimate | Feature Configuration | Measuring Component Dates | Measurement Data | Start of Estimation Horizon |
|---|---|---|---|---|
| Not Configured | Configured | Set | Any | Horizon will begin based on the most recent of the following:<br>• Process date/time less the hours configured on the feature configuration<br>• **Last Contiguous Measurement Date/Time** |

# Testing Periodic Estimation on Small Data Sets

This topic provides guidelines for testing periodic estimation on small data sets.

### How to Override the System Date

The system or process date is a key driver in the periodic estimation process. It will determine the end date/time of the estimation horizon (along with the periodic estimation configuration). When testing a single device this override can ensure that estimations never go further in time then you would like if you are working with an older set of test data.

There are two methods to override the system date (both use the format YYYY-MM-DD) that will work for the methods being described in this section.

- The **System Override Date** option on the **General System Configuration** feature configuration: this impacts all users and batches
- The **System Override Date** characteristic for a specific user: this impacts only processes executed by a particular user

# Testing Periodic Estimation on Large Data Sets

This topic provides guidelines for testing periodic estimation on large data sets.

Before running periodic estimation on a large data set it is important to have an idea of roughly how many devices and for how long you expect to estimate. This can be gauged fairly quickly by looking at the **Latest Measurement Date/Time** on the measuring component. Here is a sample query that will provide a count of measuring components by the number of months from a supplied process date the last contiguous measurement date/time is:

```
select count(*)
, round((to_date(:processDateTime,'YYYY-MM-DD-HH24.MI.SS') -
latest_msrmt_dttm)/31,0) MONTHS
from d1_measr_comp mc
group by round((to_date(:processDateTime,'YYYY-MM-DD-HH24.MI.SS') -
latest_msrmt_dttm)/31,0)
order by 2
;
```


Copyright © 2026, Oracle and/or its affiliates.

Below is a more complex query that will show you by thread how many measuring components for a given **Latest Measurement Date/Time** you will process, it does a bit more work to look at only installed devices and measuring components that are configured for periodic estimation and the measuring component is actually ready to estimate based on that configuration.

The parameters for this are:

- threads: enter 1 to avoid splitting the results into threads
- processDateTime: leave blank to use the system date, otherwise provide a date/time in the format: YYYY-MM-DD-HH24.MI.SS

```
with threads as (
 select rownum thread_nbr
 , lpad(round((rownum - 1) * (rpad('9',12,'9')+1)/:threads,0),12,'0') low_id
 , lpad(round(rownum * ((rpad('9',12,'9')+1)/:threads),0)-1,12,'0') high_id
 from dual
 connect by rownum <= :threads
)
select /*+ parallel */ threads.thread_nbr
, threads.low_Id
, threads.high_id
, nvl(mc.adj_latest_msrmt_dttm,mc.latest_msrmt_dttm) LATEST_MSRMT_DTTM
, count(*) MC_COUNT
from threads
inner join d1_dvc dvc
 on dvc.d1_device_id between threads.low_id and threads.high_id
inner join d1_dvc_cfg dc
 on dc.d1_device_id = dvc.d1_device_id
inner join d1_install_evt ie
 on ie.device_config_id = dc.device_config_id
 and ie.d1_install_dttm < sysdate
 and (ie.d1_removal_dttm is null or ie.d1_removal_dttm >= sysdate)
inner join d1_measr_comp mc
 on mc.device_config_id = dc.device_config_id
inner join d1_measr_comp_type mct
 on mct.measr_comp_type_cd = mc.measr_comp_type_cd
 and hrs_wait_before_est > 0
where (to_date(nvl(:processDateTime,sysdate),'YYYY-MM-DD-HH24.MI.SS') -
latest_msrmt_dttm) * 24 > mct.hrs_wait_before_est
group by threads.thread_nbr
, threads.low_Id
, threads.high_id
, nvl(mc.adj_latest_msrmt_dttm,mc.latest_msrmt_dttm)
order by 1, 4
;
```

# Integrations

- External Applications
- Business Flags
- Oracle Utilities Customer Care and Billing


Copyright © 2026, Oracle and/or its affiliates.

- Oracle Utilities Operational Device Management
- Oracle Utilities Analytics
- Analytics Integrations
- Oracle Utilities Analytics Visualization
- Oracle Utilities Network Management System
- Oracle Utilities Customer Self Service
- Oracle Utilities Dataconnect / Opower
- Oracle Utilities Data Intelligence
- Message Formats
- Data Import - Message Driven Bean Configuration
- Creating Application Links in External Applications

# External Applications

- Understanding External Applications
- Configuring External Applications
- Initiating Smart Meter Commands from an External Application

## Understanding External Applications

External applications are applications and systems that are external to the Oracle Utilities meter data products, and can include customer information systems such as Oracle Utilities Customer Care and Billing, outage management systems such as Oracle Utilities Network Management System, or other types of applications.

External system service providers utilize processing methods to specify how the system sends and creates data used by the two applications. For example, when Oracle Utilities Meter Data Management is integrated with Oracle Utilities Customer Care and Billing, an external system representing Oracle Utilities Customer Care and Billing would specify how usage requests are received and processed by Oracle Utilities Meter Data Management.

Refer to Understanding Process Methods for more information about processing methods.

**External Applications Impact Data Import and Export**

External Applications are configured to identify how a particular external system communicates data with Oracle Utilities Meter Data Management. This includes:

- The identifier type used to locate devices and measuring components. These are used both on import and export of data.
- The date/time format used in various data imports (i.e. whether or not the date/time format includes time zone information).

Please refer to the embedded help for more information about these fields.

Each external application can be associated to an external system which is used to define the messages that can be sent to that service provider and how each message is sent.


Copyright © 2026, Oracle and/or its affiliates.

# Configuring External Applications

This portal is used to display and maintain External Applications.

Refer to Understanding External Applications for more information.

You can access the portal from the **Admin**, then **Integration**, then **External Application**.

The following zones may appear as part of the portal's **Main** tab page:

- **External Applications List:** This zone lists all External Application records. Broadcast a record to display the details of the selected record.
- **External Application:** This zone provides information about the selected External Application.
- **Processing Method List:** This zone provides the list of processing methods defined for the External Application.
- **Translation Method List:** This zone provides the list of translation methods defined for the External Application.
- **Inbound BOs Send By Service Provider:** This zone lists inbound Business Objects that are sent by this External Application. The identification is driven by the Business Object having a Business Object Option of type "Sent By Service Provider" that references the current External Application.

# Initiating Smart Meter Commands from an External Application

Though often initiated as part of an orchestration activity (such as Enable Service or Exchange Meter) or from the Oracle Utilities Meter Data Management application, smart meter commands can also be initiated from external applications such as a customer information system such as Oracle Utilities Customer Care and Billing, a self-service application such as Oracle Utilities Digital Self-Service Transactions, or an outage management system such as Oracle Utilities Network Management System.

External applications initiate commands through the use of inbound web services, and outbound messages are used to send the results back to the initiating application. This section describes how smart meter commands can be initiated from external applications.

### Command Processing Overview

This section provides a high level overview of how the system processes commands initiated from external applications.

For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific base package objects used. Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process. This example uses an "On-Demand Read - Interval" command.

| Step | Process | Sample Object |
|---|---|---|
| 1 | External application initiates a command request via inbound web service | Inbound Web Service: D1-OnDemandReadInterval |
| 2 | Command activity is created in the Pending state. | Activity Business Object: D1-OnDemandReadInterval |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Process | Sample Object |
|---|---|---|
| 3 | The command activity goes through it's lifecycle, creating outbound communications as defined by the head end system, receiving the response. | |
| 4 | When the command enters the Completed state, an Enter algorithm sends the result to the initiating application. | Algorithm: Send On Demand Read to Edge Application (D1-SNDRDTEA) |
| 5 | The algorithm uses a "Response - Success" processing method (based on the the Response - Success (D1SR) processing role) defined for the external application to create an outbound message (based on the outbound message type defined for the processing method). | Processing Method Business Object: How to Send Activity Related O\B Messages (D1-HowToSendActivityResponse)<br>Outbound Message Business Object: Outbound Message for Communication BOs (D1-OutboundMessage) |
| 6 | The external system and message sender sends the outbound message to the external application. | |

### Inbound Web Services

This section provides information about the inbound web services available for initiating smart meter commands from an external application. The table below lists the smart meter commands available, and the specific inbound web services provided for each.

| Command | Inbound Web Service |
|---|---|
| Device Status Check | Device Status Check (D1-DeviceStatusCheck) |
| On-Demand Read (Interval) | On Demand Read Interval (D1-OnDemandReadInterval) |
| On-Demand Read (Scalar) | On Demand Read Scalar (D1-OnDemandReadScalar) |
| Remote Connect | Remote Connect (D1-RemoteConnect) |
| Remote Disconnect | Remote Disconnect (D1-RemoteDisconnect) |

See **Inbound Web Services** for more information about using inbound web services.

### Required Data Elements

The inbound web service calls used to initiate commands must include some specific data elements in order to be successfully processed.

The table below lists the specific data elements required when initiating commands via inbound web service calls. Use the **WSDL** link to view the full web service description language definition for each command.


Copyright © 2026, Oracle and/or its affiliates.

| Data Element | Description | Used by |
|---|---|---|
| `requester` | The system or person submitting the command request. This should be the name of the external application. | Device Status Check<br>On-Demand Read (Interval)<br>On-Demand Read (Scalar)<br>Remote Connect<br>Remote Disconnect |
| `utilityDeviceIdentifierNumber` | The ID of the device in the external system. This should be based on the **Utility Device ID Type** defined for the external application. For example, if the external application uses Serial Numbers for their device IDs, this would be the device's serial number. | Device Status Check<br>On-Demand Read (Interval)<br>On-Demand Read (Scalar)<br>Remote Connect<br>Remote Disconnect |
| `measurementDestination` | The destination for measurements returned by On-Demand read commands, from the Measurement Destination (MSRMT_DEST_FLG) lookup. Valid values include:<br>• Calling Edge Application (D1EA)<br>• Calling & Subscribing Edge Application (D1ES)<br>• Measurement Subscribing Edge Application (D1SE) | On-Demand Read (Interval)<br>On-Demand Read (Scalar) |

### Send to Edge Application Algorithms

As noted above, when the command enters the Completed state, an Enter algorithm sends the result to the initiating application. The table below lists the specific algorithms used with each command.

| Activity Business Object | Send to Edge Application Algorithm |
|---|---|
| Device Status Check (D1-DeviceStatusCheck) | Send Device Status Check Notification to Edge Application (D1-SNDDVSTCK) |
| On Demand Read Interval (D1-OnDemandReadInterval) | Send On Demand Read to Edge Application (D1-SNDRDTEA) |
| On Demand Read Scalar (D1-OnDemandReadScalar) | Send On Demand Read to Edge Application (D1-SNDRDTEA) |
| Remote Connect (D1-RemoteConnect) | Send Remote Connect Notification to Edge Application (D1-SRCNTEA) |
| Remote Disconnect (D1-RemoteDisconnect) | Send Remote Disconnect Notification to Edge Application (D1-SRDNTEA) |

### Sending Messages Back to the External Application

Sending outbound communications to am external application involves configuring a number of entities. This section provides an overview of the objects required to send outbound communications to external applications.

### Outbound Message Types


Copyright © 2026, Oracle and/or its affiliates.

Outbound message types define the details of outbound messages sent to the external application. You should create an outbound message type for each different type of message to be sent to the external application.

See **Outbound Messages** and **Defining Outbound Message Types** for more information about creating outbound message types.

### Message Senders

Message senders define details for sending messages to external applications. You must create a message sender for each type of message to be sent to the external application.

Note: The Message Class MUST be set to **SOAPSNDR** (Sender for real-time HTTP / SOAP messages).

See **Message Senders** for more information about creating message senders.

### External Systems

You must create an external system that represents the external application and defines the outbound message types that can be sent to the external application. The external system will be referenced on the external application.

You must configure an outbound message type on the external system for each type of message you will send to the external application. An external system's outbound message types are defined by the following:

- **Outbound Message Type**: The outbound message type (see above) created for the message
- **Processing Method**: The method by which the message will be sent. This should be set to "Real-time".
- **Message Sender**: The message sender (see above) created for the message.
- **Message XSL / Response XSL**: XSL files used to transform your message and the response to the message into the appropriate format used by the external application (Message XSL) and Smart Grid Gateway (Response XSL).

See **External Systems** for more information about creating external systems.

### External Applications

The external applications that can initiate smart meter commands must be defined as External Applications. Examples can include:

- A customer information system (such as Oracle Utilities Customer Care and Billing)
- A self-service application (such as Oracle Utilities Digital Self-Service Transactions)

Information defined for external applications include:

- **External System**: The external system (see above) the defines the types of outbound messages to be sent to the external application.
- **Our Name/ID in Their System**: The value that the external application uses to identify our system.
- **Utility Device ID Type**: The Device ID Type that will be used when communicating with the external application and it will be the assumed Device ID Type for any device identifiers sent from the external application.


Copyright © 2026, Oracle and/or its affiliates.

- **Utility Service Point ID Type**: The Service Point ID Type that will be used when communicating with the external application and it will be the assumed Service Point ID Type for any service point identifiers sent from the external application.
- **Processing Methods**: The external application's processing methods define how data relevant to the application is sent and/or created. At least one processing method should reference the outbound message type used to send responses to the external application. External applications that initiate smart meter commands should use the following processing roles:
  - Response- Success: Used to send an acknowledgement that the command was successfully executed and the results of the command, if applicable.

Refer to **Understanding External Applications** and **Configuring External Applications** for more information about external applications.

# Business Flags


- Understanding Business Flags
- Configuring Business Flags


## Understanding Business Flags

This product has implemented business flags that represent situations that exist at a customer's service point that are shared across external applications. The situations that a business flag can represent ranges from critical knowledge that requires manual analysis for resolution to purely informational notifications.

This sharing of information allows users of each system to quickly understand the status of a service point as it relates to situations that users of the system would benefit from. For example, an analytics system may identify potential theft situations that should be investigated prior to sending any usage transactions.

The subsequent sections describe business flag functionality that is specific to the service point business flag as implemented by this product.

### Business Flag Types

The system includes the following options for creating new business flag types:

| Name | Details | Business Object |
|---|---|---|
| Service Point Business Flag Type | Provides configuration for business flags associated to a service point. This type offers an ability to hold a business flag for manual analysis. | D1-SPBusinessFlagType |
| Service Point Monitor Business Flag Type | Provides configuration for business flags that should initiate service issue monitors and convey results of a field activity to the intended subscribers. | D1-SPMonitorBusinessFlagType |

Refer to About Business Flags in the *Oracle Utilities Framework Administration User Guide* for detailed information about the components of business flags.



Copyright © 2026, Oracle and/or its affiliates.

### Business Flag Business Objects

The system includes the following business flag business objects:

| Business Object | Details |
|---|---|
| Revenue Protection Insight (D1-RevenueProtectionInsight) | Business flag used to capture revenue protection related insights and the context data associated to those insights. |
| Service Point Business Flag (D1-SPBusinessFlag) | Business flags associated to a service point. |
| Initiate SP Monitoring Business Flag (D1-SPMonitorBusinessFlag) | Business flags that should initiate service issue monitors and convey results of a field activity to the intended subscribers. |

### Important Business Flag Business Object Options

Below are a list of business object options that are defined on the business flag business object associated to a given business flag type:

- **Valid Action**: identifies a BPA script that can be used to guide the user though an action that can be taken on a business flag. Note: this option should be paired with a business object lifecycle option of "Execution of Valid Actions Permitted" with a value of "true" for any lifecycle status that should support the execution of valid actions.

More detail about these options can be found by visiting a measuring component business object and inspecting the business object options.

### Field Activities can Determine Business Flag Confidence

When a business flag results in a field activity being issued and a field team is sent to investigate the service point the result of that field activity investigation will determine the confidence of the business flag. This is done through the field remarks left by the field team. The field activity remarks can either be defined as confirming the business flag or rejecting it. By selecting the appropriate field activity remark the field team will conclude the investigation into the business flag and the final confidence will be updated as appropriate.

For those field activity remarks that should either confirm or reject the business flag the Business Flag Confidence Update algorithm type should be provided for the field activity remark - activation plug-in spot.

Refer to algorithm type **Assign Business Flag Final Confidence** (D1-ABFFC) for additional details.

## Configuring Business Flags

For information on configuring business flags, see Setting Up Business Flag Configuration in the Oracle Utilities Application Framework *Administrative User Guide*.

# Oracle Utilities Customer Care and Billing


- Overview
- Configuring Master Data Synchronization
- Configuring the Bill Determinant Interface




Copyright © 2026, Oracle and/or its affiliates.

# Overview

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with a customer information system. In an integration between Oracle Utilities Meter Data Management (MDM) and a customer information system such as Oracle Utilities Customer Care and Billing (CC&B):

- Oracle Utilities Meter Data Management is typically the "system of record" for meter-related data, including meter records, meter configurations, validation, editing, and estimation (VEE) rules, bill determinant calculation rules, usage data, and calculated bill determinants.
- Oracle Utilities Customer Care and Billing (the customer information system) is typically the "system of record" for account related and service point-related data, including the rates and tariffs used to calculate bills for each account and customer.

Given this breakdown of data between the two systems, any integration between them must account for the passage of data between the two to ensure that each system can accurately perform its business functions.

# Configuring Master Data Synchronization

In most integrations with Oracle Utilities Customer Care and Billing (or other CIS), Oracle Utilities Meter Data Management is not used as the system of record for account, customer, or service point-related data. Synchronizing this data between the two systems ensures that all account, customer, and service point-related data in Oracle Utilities Meter Data Management is correct and up to date before usage transaction calculations are performed. This synchronization process is supported through a set of business objects, master configurations, batch controls, and pre-configured Inbound Web Services.

- **Initial Synchronization Requests:** Initial synchronization requests are used when initially setting up Oracle Utilities Meter Data Management. They facilitate import of data that creates devices, device configurations, measuring components, service points, install events, contacts, and usage subscriptions in Oracle Utilities Meter Data Management based on corresponding data in Oracle Utilities Customer Care and Billing.
- **Ongoing Synchronization Requests:** Ongoing synchronization requests are used when updating existing data in Oracle Utilities Meter Data Management based on changes in corresponding data in Oracle Utilities Customer Care and Billing. Ongoing synchronization requests can be used to update contacts, devices, device configurations, measuring components, install events, service points, and usage subscriptions.
- **Composite Synchronization Requests:** Composite synchronization requests are requests that contain synchronization requests for multiple types of data within a single request. For example, a composite request could contain requests to update device, device configuration, measuring component, and install event data. This supports situations where multiple types of data must be updated based on a single change in Oracle Utilities Customer Care and Billing.

### Master Configuration

The following master configurations are used in Oracle Utilities Meter Data Management to configure the sync process between CIS and MDM:

| Master Configuration | Name / Description |
|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | |
|---|---|
| Master Data Synchronization Configuration | Lists all foreign key references that need resolution. Each one should reference the view that contains the external key / production key cross-reference. For entities that undergo both the initial and the ongoing sync, two views are specified. For entities that undergo the ongoing sync, an external system / ID type mapping is specified to cater for entities that might be synchronizing from more than one external system. |
| Seeder Sync Request Master Configuration | Lists the maintenance objects (device, device configuration, etc.) that require synchronization. Each references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified. |

### Inbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing uses the following inbound (to Oracle Utilities Mater Data Management) synchronization business objects:

| Object Name | Business Object | Description |
|---|---|---|
| Device Config Composite Sync Request | D1-CompositeSyncRequestDC | A composite synchronization request that handles syncing in Device Configuration and creating related Measuring Component. This business object a child of the D1-CompositeSyncRequest business object. |
| Contact Initial Sync Request | D1-InitialSyncRequestContact | Instances of this business object represent individual initial contact synchronization requests. |
| Device Configuration Initial Sync Request | D1-InitialSyncRequestDC | Instances of this business object represent individual initial device configuration synchronization requests. |
| Device Initial Sync Request | D1-InitialSyncRequestDevice | Instances of this business object represent individual initial device synchronization requests. |
| Install Event Initial Sync Request | D1-InitialSyncRequestIE | Instances of this business object represent individual initial install event synchronization requests. |
| Measuring Component Initial Sync Request | D1-InitialSyncRequestMC | Instances of this business object represent individual initial measuring components synchronization requests. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Service Point Initial Sync Request | D1-InitialSyncRequestSP | Instances of this business object represent individual initial service point synchronization requests. |
| Dynamic Option Initial Sync Request | D2-InitialSyncRequestDynOpt | Instances of this business object represent individual initial dynamic option synchronization requests. |
| Dynamic Option Event Initial Sync Request | D2-InitialSyncRequestDynOptEvt | Instances of this business object represent individual initial dynamic option event synchronization requests. |
| Usage Subscription Initial Sync Request | D2-InitialSyncRequestUS | Instances of this business object represent individual initial usage subscription synchronization requests. |
| Ongoing Sync Request Acknowledgement | D1-OngoingSyncReqAckMsg | Used to send a message to the sending system to acknowledge receipt of an ongoing synchronization request. |
| Scalar Meter Read Ongoing Sync Request | D1-OngoingSyncReqScalarMtrRead | Instances of this business object represent individual ongoing scalar meter read synchronization requests. |
| Contact Ongoing Sync Request | D1-OngoingSyncRequestContact | Instances of this business object represent individual ongoing contact synchronization requests. |
| Device Configuration Ongoing Sync Request | D1-OngoingSyncRequestDC | Instances of this business object represent individual ongoing device configuration synchronization requests. |
| Device Ongoing Sync Request | D1-OngoingSyncRequestDevice | Instances of this business object represent individual ongoing device synchronization requests. |
| Install Event Ongoing Sync Request | D1-OngoingSyncRequestIE | Instances of this business object represent individual ongoing install event synchronization requests. |
| Measuring Component Ongoing Sync Request | D1-OngoingSyncRequestMC | Instances of this business object represent individual ongoing measuring component synchronization requests. |
| Service Point Ongoing Sync Request | D1-OngoingSyncRequestSP | Instances of this business object represent individual ongoing service point synchronization requests. |
| Dynamic Option Ongoing Sync Request | D2-OngoingSyncRequestDynOpt | Instances of this business object represent individual ongoing dynamic option synchronization requests. |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| Dynamic Option Event Ongoing Sync Request | D2-OngoingSyncRequestDynOptEvt | Instances of this business object represent individual ongoing dynamic option event synchronization requests. |
| Usage Subscription Ongoing Sync Request | D2-OngoingSyncRequestUS | Instances of this business object represent individual ongoing usage subscription synchronization requests. |
| Sync Request Seeder | D1-SyncRequestSeeder | Used to identify the appropriate synchronization business object to create when processing synchronization requests. |
| Contact Synchronization Add | D1-SynchronizationAddContact | Used when adding a new contact as a result of a synchronization request. |
| Device Configuration Synchronization Add | D1-SynchronizationAddDC | Used when adding a new device configuration as a result of a synchronization request. |
| Device Synchronization Add | D1-SynchronizationAddDevice | Used when adding a new device as a result of a synchronization request. |
| Install Event Synchronization Add | D1-SynchronizationAddIE | Used when adding a new install event as a result of a synchronization request. |
| Measuring Component Synchronization Add | D1-SynchronizationAddMC | Used when adding a new measuring component as a result of a synchronization request. |
| Service Point Synchronization Add | D1-SynchronizationAddSP | Used when adding a new service point as a result of a synchronization request. |
| Usage Subscription Synchronization Add | D2-SynchronizationAddUS | Used when adding a new usage subscription as a result of a synchronization request. |

### Batch Controls

Batch controls perform processing for initial synchronization requests such as allocating keys to data, resolving foreign keys, and loading data (instantiating business objects representing entities such devices, measuring components, etc.).

"Initial Sync Request - Resolve Keys XXX" batch controls invoke a generic maintenance object transition process to invoke the "Resolve Keys - Initial Sync" algorithm for synchronization requests of the appropriate type. Parameters used by "resolve keys" batch controls include:

- **Maintenance Object**: (Required) the maintenance object (device, device configuration, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (device for device synchronization requests, service point for service point synchronization requests, etc.)
- **Restrict By Batch Code**: Restricts processing to synchronization requests whose current state is linked to this batch code.
- **Restrict By Business Object**: Restricts processing to synchronization requests linked to this business object.


Copyright © 2026, Oracle and/or its affiliates.

- **Restrict By Status Code**: Restricts processing to synchronization requests of this status (default: KEY_ALLOCATD).
- **Max Errors**: Specifies the maximum number of errors allowed before the process exits.

"Initial Sync Request - Load Data XXX" batch controls load data (created new instances of business objects) for requests of the appropriate type (device, measuring component, etc,). Parameters used by "load data" batch controls include:

- **Maintenance Object**: (Required) the maintenance object (device, device configuration, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (device for device synchronization requests, service point for service point synchronization requests, etc.)
- **Restrict By Batch Code**: Restricts processing to synchronization requests whose current state is linked to this batch code.
- **Restrict By Business Object**: Restricts processing to synchronization requests linked to this business object.
- **Max Errors**: Specifies the maximum number of errors allowed before the process exits.

The table below lists the batch controls used by initial synchronization requests:

| Batch Code | Name / Description |
|---|---|
| D1-CMSYN | Composite Sync Request |
| D1-SIIER | Initial Sync Request - Error |
| D1-SILCN | Initial Sync Request - Load Data Contact |
| D1-SILDC | Initial Sync Request - Load Data DC |
| D1-SILDV | Initial Sync Request - Load Data Device |
| D1-SILIE | Initial Sync Request - Load Data IE |
| D1-SILMC | Initial Sync Request - Load Data MC |
| D1-SILSP | Initial Sync Request - Load Data SP |
| D1-SILUS | Initial Sync Request - Load Data US |
| D1-SIKCN | Initial Sync Request - Resolve Keys Contact |
| D1-SIKDC | Initial Sync Request - Resolve Keys DC |
| D1-SIKDV | Initial Sync Request - Resolve Keys Device |
| D1-SIKIE | Initial Sync Request - Resolve Keys IE |
| D1-SIKMC | Initial Sync Request - Resolve Keys MC |
| D1-SIKSP | Initial Sync Request - Resolve Keys SP |
| D1-SIKUS | Initial Sync Request - Resolve Keys US |
| D1-SIOER | Ongoing Sync Request -Error |
| D1-SIOPE | Ongoing Sync Request - Pending |
| D1-SRSDE | Sync Request Seeder - Error |
| D2-SIKDO | Initial Sync Request - Resolve Keys DO |
| D2-SIKDE | Initial Sync Request - Resolve Keys DOE |
| D2-SIKUS | Initial Sync Request - Resolve Keys US |
| D2-SILDO | Initial Sync Request - Load Data DO |
| D2-SILDE | Initial Sync Request - Load Data DOE |
| D2-SILUS | Initial Sync Request - Load Data US |
| F1-SAKRQ | Sync Request Allocate Keys Monitor |


Copyright © 2026, Oracle and/or its affiliates.

| | |
|---|---|
| F1-SRLRQ | Sync Request Load Records Monitor |
| F1-SYNRQ | Sync Request Monitor Process |
| F1-SYSRQ | Sync Request Sampling Monitor (Deferred) |

### Batch Control Scheduling

The following table specifies the order in which the batch controls on the Initial Sync Request BO life cycle should be executed. The first row identifies the maintenance object for which the synchronization request is intended and the first column specifies the type of process.

| | Contact | Usage Subscription | Service Point | Install Event | Device Configuration | Device | Measuring Component | Dynamic Option | Dynamic Option Event |
|---|---|---|---|---|---|---|---|---|---|
| Transformation / Schema Validation Job | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Key Allocation Job | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Foreign Key Resolution / BO Validation Job (dependent on ALL Key Allocation Jobs finishing) | 3 | 8 | 3 | 6 | 3 | 3 | 6 | 3 | 5 |
| Load Job | 4 | 9 | 5 | 7 | 5 | 4 | 7 | 4 | 6 |

Note that before the Key Resolution job is run, all the Key Allocation Jobs need to finish. This ensures that all foreign key references can be subsequently resolved.

Some business object-level validation is dependent on other entities being completely loaded first. The sequence numbers above allow for this. For example, usage subscriptions business object validation is dependent on service points existing; Install Event business object validation is dependent on both service points and devices existing.

### Inbound Web Services

Inbound web services are used to facilitate invoking the Sync Request Seeder business object by the middleware components upon receipt of a synchronization request.

The table below lists the pre-configured Inbound Web Services used to process synchronization requests sent from Oracle Utilities Customer Care and Billing.


Copyright © 2026, Oracle and/or its affiliates.

| Inbound Web Service | Description | Schema Name |
|---|---|---|
| D1-SyncRequestInbound | Sync Request Inbound | D1-SyncRequestSeeder (BO) |
| D1-SyncRequestInboundComposite | Sync Request Inbound Composite | D1-CompositeSyncRequestDC (BO) |

### Example Sync Process Steps

This section provides an overview of the processing that takes place when a synchronization request is sent. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects involved.

| Step | Process | Objects |
|---|---|---|
| 1 | Oracle Utilities Customer Care and Billing sends a synchronization request to the middleware integration layer.<br>For example, consider a request to update information about a service point. | |
| 2 | The middleware components transforms the request from the Customer Care and Billing format, to the format used by Oracle Utilities Meter Data Management (this format is based on the business object schemas of the synchronization request business objects). | |
| 3 | The middleware component invoke the appropriate Inbound Web Service, and sends the transformed request. | Inbound Web Service:<br>D1- SyncRequestInbound (mapped to the D1-SynrRequestSeeder business object) |
| 4 | The Inbound Web Service invokes the Sync Request Seeder business object, which in turn, determines which synchronization request business object to create (based on the type of data in the synchronization request and the Seeder Sync Master Configuration). | Synchronization Request BO:<br>D1- OngoingSyncRequestSP |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| 5 | For initial synchronization requests, background processing creates master data for each synchronization request, including the following steps:<br>• Data Transformation / Schema Validation<br>• Allocate Keys<br>• Resolve Foreign Keys / Validate Business Object<br>• Load Data | |
| 6 | For ongoing synchronization requests, the following steps are performed by Enter algorithms on the synchronization request business object lifecycle states:<br>• Data Transformed/Basic Schema Validated:<br>• Determine Sync Request BO<br>• Setup Transformed Data<br>• Pre-Added:<br>• Sync Request Pre-Add Data<br>• FKs Resolved:<br>• Resolve Keys - Ongoing Sync<br>• Updating:<br>• Sync Request Update Data | Setup Transformed Data Algorithm:<br>D1- SETTRANDT<br>Sync Request Pre-Add Data Algorithm:<br>D1-SR-PREADD<br>Resolve Keys - Ongoing Sync Algorithm:<br>D1-RESKEYFAL<br>Sync Request Update Data Algorithm:<br>D1- SR-UPDDAT |

# Configuring the Bill Determinant Interface

Oracle Utilities Customer Care and Billing uses bill determinant data (usage transactions based on meter readings) calculated and stored in Oracle Utilities Meter Data Management when calculating bills for customers. This process allows Oracle Utilities Customer Care and Billing to send requests for usage transaction calculations to Oracle Utilities Meter Data Management, which in turn performs the requested calculations, and publishes the results back to Oracle Utilities Customer Care and Billing. Processing usage transaction requests is supported through a set business objects, pre-configured Inbound web services, and processing methods.

### Business Objects

The table below lists the business objects used when processing usage transaction requests.

| Business Object | Description |
|---|---|
| D2-UsgTranSeeder | Used to determine the usage transaction business object to use when creating new usage transactions. |


Copyright © 2026, Oracle and/or its affiliates.

### Inbound Web Services

Inbound web services are used to facilitate invoking the Usage Transaction Seeder business object by the middleware components upon receipt of a usage transaction request, and to update the usage transaction upon completion of the bill.

The table below lists the pre-configured Inbound Web Services used to process usage transaction requests sent from Oracle Utilities Customer Care and Billing.

| Inbound Web Service | Description | Schema Name |
|---|---|---|
| D2-UsageTransactionRequestInbound | Usage Transaction Request Inbound | D2-UsgTranSeeder (BO) |
| D2-UsageTransactionUpdateInbound | Update Usage Transaction Bill Flags | D2-UpdUTMul (Service Script) |

### Service Provider Processing Methods

Processing methods are used to determine the usage transaction business object to use when creating usage transactions based on the requests, and for determining the method by which usage transactions are sent back to Oracle Utilities Customer Care and Billing.

| Processing Method | Business Object | Description |
|---|---|---|
| How To Send US Information - Online | D2-HowToSendUSInfoOnline | Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems in real time. |
| How To Send US Related Information | D2-HowToSendUSInformation | How To Send US Related Information Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems. |
| How To Create US Related Information | D2-HowToCreateUSInformation | Used to determine the type of usage transaction business object to create. |
| How To Send US Related Information Batch | D2-HowToSendUSInfoBatch | Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems via batch processing. |

### Example Bill Determinant Process Steps

This section provides an overview of the processing that takes place when a usage transaction request is sent. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects involved.

| Step | Process | Objects |
|---|---|---|


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| 1 | Oracle Utilities Customer Care and Billing sends a usage transaction request to the middleware integration layer. | |
| 2 | The middleware components transforms the request from the Customer Care and Billing format, to the format used by Oracle Utilities Meter Data Management (this format is based on the business object schemas of the synchronization request business objects). | |
| 3 | The middleware component invoke the appropriate Inbound Web Service, and sends the transformed request. | Inbound Web Service:<br>D2-UsageTransactionRequestInbound (mapped to the D2-UsgTranSeeder business object) |
| 4 | The Inbound Web Service invokes the Usage Transaction Seeder business object. This business object:<br>• Determines the usage subscription ID based on an external usage subscription ID<br>• Determines the appropriate usage transaction business object to create | Processing Method:<br>D2-HowToCreateUSInformation (How To Create Usage Subscription Related Information) |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| 5 | The usage transaction determines the usage calculation group(s) to use when calculating usage. These are retrieved from the usage subscription (with a fallback to the usage subscription type).<br>This base package logic can be overridden by specifying an algorithm in the **Usage Calculation Group Determination Override Algorithm** field on the usage subscription type. If an algorithm is specified in this field, its logic overrides the existing method of determining the usage calculation group.<br>A base package algorithm (D2-DRVUSGGRP) is provided that can be used here. This algorithm uses the **CCB Rate Schedule** extendable lookup used to map the rate schedules in Oracle Utilities Customer Care and Billing to usage calculation groups. The algorithm's logic looks for a combination of rate history list entries on the usage subscription and the type of device configuration installed during the bill period to determine the usage calculation group(s) to use for bill determinants calculation. | |
| 6 | The usage transaction business object calculates usage based on the date range provided in the request. | |
| 7 | If the usage transaction has a sub usage transactions, it checks the status of each. | |
| 8 | If the usage transaction is configured to require an approval, a To Do Entry is created. | |
| 9 | The usage transaction then sends the usage transaction to any subscribing systems. | Processing Method:<br>D2- HowToSendUSInformation (How To Send US Related Information) |


Copyright © 2026, Oracle and/or its affiliates.

| | | |
|---|---|---|
| 10 | Upon completion of the bill, Oracle Utilities Customer Care and Billing invokes an inbound web service to update the "Used on Bill" and/or "Linked to Frozen Bill Segment" fields as appropriate to indicate if the usage transaction was used on a bill. | Inbound Web Service:<br>D2-UsageTransactionUpdateInbound<br>(Update Usage Transaction Bill Flags) |

# Oracle Utilities Operational Device Management


- Overview
- Configuring Master Data Synchronization


## Overview

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with Oracle Utilities Operational Device Management. In an integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management:

- Oracle Utilities Meter Data Management is typically considered the "system of record" for service points (asset locations), contacts, and device installation information (install events)
- Oracle Utilities Operational Device Management is typically considered the "system of record" for assets (devices) and asset specification configuration (device configurations)

Given this breakdown of data between the two systems, any integration between them must account for the passage of data between the two to ensure that each system can accurately perform its business functions. The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management is based on data synchronization between the two systems.

## Configuring Master Data Synchronization

The specific data synchronization flows supported between Oracle Utilities Meter Data Management (MDM) and Oracle Utilities Operational Device Management (ODM) include the following:

- **Asset-Device Synchronization**: As new assets and asset specification configurations are created or changed in ODM, corresponding devices and device configurations must be created or changed in MDM
- **Service Point/Contact - Asset Location/Contact**: As Service Points and/or Contacts are created or changed in MDM, corresponding Asset Locations and Contacts must be created or changed in ODM
- **Install Events- Asset Location/Disposition**: As devices are installed/removed in MDM, corresponding changes to an asset's Disposition (location and status) must be made in ODM

This synchronization process is supported through a set of business objects, master configurations, batch controls, and pre-configured Inbound Web Services. Refer to the *Oracle*



Copyright © 2026, Oracle and/or its affiliates.

*Utilities Integration for Device Operations Implementation Guide* for more information about this integration and the data synchronization processes used by this integration.

### Master Configuration

The following master configurations are used to configure the sync process between MDM and ODM:

| Master Configuration | Name / Description |
|---|---|
| Master Data Synchronization Configuration | Lists all foreign key references that need resolution. Each one should reference the view that contains the external key / production key cross-reference. For entities that undergo both the initial and the ongoing sync, two views are specified. For entities that undergo the ongoing sync, an external system / ID type mapping is specified to cater for entities that might be synchronizing from more than one external system. |
| Seeder Sync Request Master Configuration | Lists the maintenance objects (device, device configuration, etc.) that require synchronization. Each references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified. |
| ODM Integration Master Configuration | Specifies the external system used to represent Oracle Utilities Operational Management, and the URL for the Oracle Utilities Operational Management application.<br>Also specifies the outbound message types used to send synchronization requests to Oracle Utilities Operational Management. |

### Inbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following inbound (to Oracle Utilities Mater Data Management) synchronization business objects:

| Object Name | Business Object | Description |
|---|---|---|
| Device Ongoing Sync Request | D1-OngoingSyncRequestDevice | Used to synchronize devices and device configurations in Oracle Utilities Mater Data Management based on assets and asset specification configurations created/updates in Oracle Utilities Operational Device Management. |

### Outbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following outbound (from Oracle Utilities Mater Data


Copyright © 2026, Oracle and/or its affiliates.

Management) synchronization business objects (based on the F1-SYNC REQ maintenance object)

| Object Name | Business Object | Description |
|---|---|---|
| ODM Sync Request | D1-ODMContactSyncRequest | Used to synchronize contacts in Oracle Utilities Operational Device Management based on contacts created/updated in Oracle Utilities Meter Data Management |
| ODM Install Event Sync Request | D1-ODMInstallEventSyncRequest | Used to synchronize asset location/disposition in Oracle Utilities Operational Device Management based on install events created/ updated in Oracle Utilities Meter Data Management |
| ODM SP Sync Request | D1-ODMSPSyncRequest | Used to synchronize asset locations in Oracle Utilities Operational Device Management based on service points created/updated in Oracle Utilities Meter Data Management. This includes updating asset locations when equipment is attached to or remove from a service point. |

# Oracle Utilities Analytics


- Overview
- Configuring Extracts


## Overview

Oracle Utilities Meter Data Management can also be integrated with Oracle Utilities Analytics (OUA) to allow users to view analytic data based on usage, events, and other data tracked in Oracle Utilities Meter Data Management.

Oracle Utilities Meter Data Management Analytics comprises the following products:

- **Oracle Utilities Meter Data Analytics**: dashboards, dashboard pages, and analytics used to view usage, event, and other data from Oracle Utilities Meter Data Management
- **Oracle Utilities Meter Data Management Extractors and Schema**: the star schema and extract programs used by Oracle Utilities Meter Data Analytics.

## Configuring Extracts

Oracle Utilities Meter Data Management includes the following components used when integrating with Oracle Utilities Analytics:



Copyright © 2026, Oracle and/or its affiliates.

### Master Configuration

The following master configurations are used to configure the sync process between MDM and OUA:

| Master Configuration | Description |
|---|---|
| Analytics Master Configuration - ODI-Based | This master configuration is primarily used to configure the information that will be used in the extraction of data for business intelligence when using the Oracle Data Integrator extract, transform, and load (ETL) process. |

### Batch Controls

| Batch Code | Name / Description |
|---|---|
| D1-SPSDL | Service Point Snapshot Download |
| D2-DUGDL | Usage Subscription Usage Calculation Group Download |
| D2-USGDL | Service Point Usage Snapshot Download |
| D2-UUSDL | SP Unreported Usage Snapshot Download |
| D2-VEEDL | SP VEE Exception Snapshot Download |

Refer to the following documentation for more information about Oracle Utilities Analytics:

*Oracle Utilities Extractors and Schema for Oracle Utilities Meter Data Management Data Mapping Guide*

*Oracle Utilities Analytics Dashboards for Meter Data Analytics Metric Reference Guide*

*Oracle Utilities Analytics Admin Guide*

# Analytics Integrations

This section describes integrations with external analytics applications such as Oracle Utilities Data Intelligence.

- Business Flag Integration
- Meter Insights and Investigations
- Usage and Event Integration

## Business Flag Integration

With the business flag integration with external analytics applications:

- Either Oracle Utilities Meter Solution Cloud Service/Meter Data Management or an external analytics application can initiate a business flag
- Oracle Utilities Meter Solution Cloud Service/Meter Data Management can notify external analytics applications of a change in status of a business flag:
  - if a business flag in error has been discarded
  - if a business flag's confidence has been updated


Copyright © 2026, Oracle and/or its affiliates.

  - with the results of field work initiated by a business flag
- Oracle Utilities Data Intelligence can also publish Meter Operations insights as service point business flags. See **Meter Operations Insights Business Flags** for insight-specific configuration and context information.

For additional details see About Business Flags in the Oracle Utilities Application Framework *Administrative User Guide*.

- Configuring Business Flag Integration
- Meter Operations Insights Business Flags

# Configuring Business Flag Integration

Oracle Utilities Meter Solution products include the following components used business flag integrations:

### Master Configuration

The following master configurations are used to configure the data synchronization process between MDM and external analytics applications:

| Master Configuration | Description |
|---|---|
| Oracle Utilities DataRaker Integration Master Configuration | This master configuration is used to enable deep linking into Oracle Utilities Analytics Insights. Multiple URL patterns are supported with an ability to provide substitution variables to satisfy the URL parameters (e.g. "&searchTech="). This also provides a mapping between Oracle Utilities Meter Data Management service types and point types defined in external analytics applications. |

### Oracle Utilities Data Intelligence

Receiving business flag insights from external analytics applications such as Oracle Utilities Data Intelligence requires set up of the following:

- Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup
- Measurement Identification - Electric Configuration (D1-MsrmtIdentifyElectricConfig) master configuration
- Measurement Identification - Gas Configuration (D1-MsrmtIdentifyGasConfig) master configuration
- Measurement Identification - Water Configuration (D1-MsrmtIdentifyWaterConfig) master configuration

See Defining Standard Measurement Identifiers for more information.

### Service Provider Processing Methods

The following processing methods should be associated to the service provider configured for external analytics applications:


Copyright © 2026, Oracle and/or its affiliates.

| Processing Role | Description |
|---|---|
| Business Flag Confidence Change | This role will identify the appropriate message to send when a business flag confidence has been changed (e.g. from Suspected to Confirmed). When not configured no message will be sent. |
| Business Flag Discarded Notification | This role will identify the appropriate message to send when a business flag has been discarded. When not configured no message will be sent. |
| Business Flag Field Work Details | This role will identify the appropriate message to send when a business flag has completed that initiated field work. When not configured no message will be sent. |
| Business Flag Added Notification | This role will identify the appropriate message to send when a business flag has been created from within Oracle Utilities Meter Data Management. When not configured no message will be sent. |

### Extendable Lookups

The following extendable lookups should be configured when setting up business flag integrations:

| Extendable Lookup | Description |
|---|---|
| Business Flag Standard Name | If external analytics applications will be communicating business flags using a non-standard name provide an entry in the Valid External Name with the non-standard name for each standard name communicated by external analytics applications. |

## Meter Operations Insights Business Flags

Meter Operations Insights are analytics produced by Oracle Utilities Data Intelligence (OUDI) that evaluate meter and service-point data to identify potential operational issues. OUDI publishes the resulting insights to Meter Solution Cloud Service or Meter Data Management as Business Flags with supporting context, so users can prioritize and review potential issues in their existing MDM workflow.

Meter Operations Insights sends the following insights as business flags:

- Meter Bypass
- Frequent Power Outage
- Power Factor Monitoring

The Business Flag identifies the service point and includes context that supports review of the insight.

For general information about business flags, see Business Flags in the *Application Framework Administrative User Guide*. For general integration configuration, see Business Flag Integration and Configuring Business Flag Integration.


Copyright © 2026, Oracle and/or its affiliates.

### Meter Operations Insight Context

The following optional context values can be included with Meter Operations business flags:

| Insight | Context Value | Description |
|---|---|---|
| Meter Bypass | Usage Before | Daily average kWh observed before the suspected bypass. |
| Meter Bypass | Usage After | Daily average kWh observed after the suspected bypass. |
| Frequent Power Outage | Power Outage Count | Number of power outages associated with the insight. |
| Power Factor Monitoring | Power Factor | Calculated power factor. |
| Power Factor Monitoring | Usage Loss | Aggregated kWh loss for the duration of the business flag. |

Context information is displayed on the business flag when it is available. A business flag can be created with or without context values.

### Configuring Meter Operations Insight Business Flags

Configure a separate Service Point Business Flag Type for each Meter Operations insight you wish to enable. Use the Service Point Business Flag (D1-SPBusinessFlag) business object unless your implementation requires monitor-driven follow-up or field work.

The following table lists the business flag types to configure. Use the standard names and any delivered type codes supplied with your release.

| Insight | Business Object | Description |
|---|---|---|
| Meter Bypass | Service Point Business Flag (D1-SPBusinessFlag) | Review the business flag using the standard Business Flag portal. |
| Frequent Power Outage | Service Point Business Flag (D1-SPBusinessFlag) | Review the business flag using the standard Business Flag portal. |
| Power Factor Monitoring | Service Point Business Flag (D1-SPBusinessFlag) | Review the business flag using the standard Business Flag portal. |

The Service Point Business Flag (D1-SPBusinessFlag) business object does not create a service issue monitor, field activity, or investigation task. If your implementation requires monitor-driven follow-up or field work, configure the applicable business flag type using the Initiate SP Monitoring Business Flag (D1-SPMonitorBusinessFlag) business object. See Service Issue Monitors and Service Order Management for configuration details.

If the external system uses a name that differs from the configured standard name, add the external name to the **Business Flag Standard Name** extendable lookup. See Configuring Business Flag Integration for more information.

### Configuring Measurement Identification

Meter Operations Insights uses the measurement identification configuration in Meter Solution Cloud Service and Meter Data Management to identify the customer-specific measurement channels used by the insights. Configure the mappings for the measurements required by the insights you enable. See Defining Standard Measurement Identifiers for more information.


Copyright © 2026, Oracle and/or its affiliates.

For Power Factor Monitoring, configure the mappings for power factor and the supporting measurements used by the implementation, including kVA when applicable. If the required power factor data is not available in the system, configure pass-through measuring processing before enabling Power Factor Monitoring. See Measuring Component Filtering for more information.

### Reviewing Meter Operations Insight Business Flags (Section)

Use the **Business Flag** portal to search for and review Meter Operations insight business flags. You can search by business flag type to review insight records across service points. Review the business flag context together with the service point information and follow the investigation process defined by your implementation.

# Meter Insights and Investigations

The Meter Insights and Investigations functionality leverages external analytics application business flag integrations to support receipt of revenue protection events (referred to as "insights") from external analytics applications for research and investigation using Oracle Utilities Meter Data Management (or Oracle Utilities Meter Solution Cloud Service). Revenue protection insights sent from external analytics applications for investigation are events that suggest potential theft or loss of revenue such as unexpected consumption from an inactive meter, evidence of meter tampering, discovery of an improperly installed meter (such as a meter installed upside down or in reverse), and so on. Insights received from external analytics applications create business flags, which in turn create related Meter Insight Investigation service tasks (referred to as “investigations”), which can then be researched and investigated using a dedicated **Meter Insights Investigations Viewer** portal.

See **Business Flags** in the *Administrative User Guide* for general information about the business flag functionality provided by the Oracle Utilities Application Framework.

See **Business Flag Integration** in the *Administrative User Guide* for more information about the Business Flag integration with external analytics applications.

See **Service Issue Monitors** in the *Business User Guide* and **Service Task Types** in the *Administrative User Guide* for general information about service issue monitors.

See **Using the Meter Insights Investigations Viewer Portal** in the *Business User Guide* for more information about viewing meter insight service issue monitors.

### Meter Insights and Investigation Business Objects

Business flags and service issue monitors used by Meter Insights and Investigations functionality are based on specific business objects:

| Type of Object | Business Object |
|---|---|
| Business Flag | Revenue Protection Insight (D1-RevenueProtectionInsight) |
| Service Issue Monitor (Service Task) | Meter Insight Investigation (D1-MeterInsightInvestigation) |
| Investigation Type (Service Task Type) | Meter Insight Investigation Type (D1-MtrInsightInvestigationType) |

### Meter Insights and Investigation Processing

The table below provides a brief description of the process by which insights are received and processed, and lists example objects used by the Oracle Utilities Meter Data Management


Copyright © 2026, Oracle and/or its affiliates.

(and Oracle Utilities Meter Solution Cloud Service). Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

| Step | Process | Example Data |
|---|---|---|
| 1. | Oracle Utilities Analytics Insights sends events ("insights") to Meter Data Management, via the Business Flag integration, creating business flags based on the business flag type configuration. | Business Flag Business Object: Revenue Protection Insight<br>(D1-RevenueProtectionInsight) |
| 2. | An Enter algorithm on the "Create Service Issue Monitor" state creates a Service Issue Monitor based on the Service Task type defined on the business flag's type.<br>The investigation is created in "Pending" status. | Enter Algorithm: Business Flag - Create Service Issue Monitor<br>(D1-CREATESIM)<br>Note: This algorithm is on the parent business object (D1-SPMonitorBusinessFlag).<br>Service Issue Monitor (Service Task) Business Object: Meter Insight Investigation (D1-MeterInsightInvestigation) |
| 3. | The investigation evaluates the service point (based on configuration of the service task type).<br>• If there are no issues, the investigation moves into the Discarded (No Investigation Needed) state.<br>• Otherwise, the investigation continues to the Open state. | Evaluate Algorithm: Meter Insight Investigation - Analyze Service Point For Service Issues<br>(D1-MII-ANZ) |
| 4. | An Enter algorithm on the Open state creates a To Do entry. | Open Enter Algorithm: Meter Insight Investigation - Create To Do Entry (D1-MIICRTODO) |
| 5. | Once the investigation is in the Open state, a business user can use the **Meter Insights Investigation Viewer** portal to examine the issue. See "Using the Meter Insights Investigation Portal" in the *Business User Guide* for more information about using this portal.<br>The user can add comments and update the investigation to one of the following states:<br>• Waiting for Field Work<br>• Issue Resolved<br>• No Issue<br>• Discarded<br>When the investigation moves into the Issue Resolved, No Issue, or Discarded states, it transitions the business flag the Completed state with a Final Confidence based on the investigation's state (see below). | |


Copyright © 2026, Oracle and/or its affiliates.

| Step | Process | Example Data |
|---|---|---|
| 6. | If the service issue monitors enters the Waiting for Field Work state, a Monitor algorithm evaluates the Completion Criteria on the Service Task Type and transitions the investigation to one of the following states:<br>• Fieldwork Error<br>• Open<br>• Issue Resolved<br>• No Issue<br>• Inconclusive<br>When the investigation moves into the Issue Resolved, No Issue, or Inconclusive states, it transitions the business flag the Completed state with a Final Confidence based on the investigation's state (see below). | Waiting For Field Work Monitor Algorithm: Evaluate Completion Criteria<br>(D1-MIIEVCMCR) |
| 7. | If the service issue monitors enters the Issue Resolved state:<br>• An Enter algorithm completes the To Do entry<br>• An Enter algorithm transitions the Business Flag to the Completed state (Final Confidence: Confirmed) | Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE)<br>Enter Algorithm: Transition Business Flags - Completed (D1-TRBUSFLGC) |
| 8. | If the service issue monitors enters the No Issue state,<br>• An Enter algorithm completes the To Do entry<br>• Transitions the Business Flag to Completed (Final Confidence: Rejected) | Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE)<br>Enter Algorithm: Transition Business Flags - Rejected (D1-TRBUSFLGR) |
| 9. | If the service issue monitors enters the Inconclusive state,<br>• An Enter algorithm completes the To Do entry<br>• Transitions the Business Flag to Completed (Final Confidence: Suspected) | Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE)<br>Enter Algorithm: Transition Business Flags - Suspected (D1-TRBUSFLGI) |
| 10. | If the service issue monitors enters the Discarded state<br>• An Enter algorithm completes the To Do entry<br>• Transitions the Business Flag to Completed (Final Confidence: Rejected) | Enter Algorithm: Generic To Do Completion on Enter (D1-TODOCMPLE)<br>Enter Algorithm: Transition Business Flags - Rejected (D1-TRBUSFLGR) |

- Configuring Meter Insights and Investigations


Copyright © 2026, Oracle and/or its affiliates.

# Configuring Meter Insights and Investigations

Configuring the Meter Insights and Investigations functionality involves creation and configuration of the following:

- Business Flag Types
- Extendable Lookup Values
- To Do Types
- Service Tasks Types
- Master Configuration

### Business Flag Types

Create a business flag type for each type of revenue protection insight that will be received from external analytics applications, including the following:

- **Business Object**: Service Point Monitor Business Flag Type (D1-SPMonitorBusinessFlagType)
- **Related Transaction BO**: Revenue Protection Insight (D1-RevenueProtectionInsight)
- **Service Issue Monitor Type**: The service task type for investigations created from business flags of this type.

> **Note**
>
> Because the **Service Issue Monitor Type** field is required, when initially creating business flag types, select a placeholder Service Issue Monitor Type to enable you to save the business flag type. After you have created the appropriate service task types, return to each business flag type and select the correct Service Issue Monitor Type.

Define other fields based on your business requirements.

### Extendable Lookups

The **Meter Insights Investigation Viewer** portal uses the following profile extendable lookups when displaying consumption and insights (events):

- **360 View Event Bar Profile**: Defines events to highlight that are pertinent to investigations, including your new business flag types
- **Final Values Overlay Profile**: Defines values to graph that are pertinent to investigations

Use the **Extendable Lookup** portal to create new extendable lookup values:

1. Create a new 360 View Event Bar Profile extendable lookup value that includes your business flag types as well as other events pertinent to your investigations.
2. Create a new Final Values Overlay Profile extendable lookup value that includes the values to graph that are pertinent to your investigations.

These extendable lookup values will be referenced by your service task types.


Copyright © 2026, Oracle and/or its affiliates.

### To Do Types

Use the **To Do Type** portal to create a To Do Type (or update an existing To Do Type) with the Set Investigation Priority (D1-STINVSGPR) algorithm defined for the **Calculate Priority** system event.

These To Do Types will be referenced by your service task types.

### Service Task Types

Create a service task type for each type of revenue protection insight (each type of investigation) that will be received from Oracle Utilities Analytics Insights, including the following:

- **Business Object**: Meter Insight Investigation Type (D1-MtrInsightInvestigationType)
- **Related Transaction BO**: Meter Insight Investigation (D1-MeterInsightInvestigation)
- **Evaluation Criteria**: The criteria used to evaluate the service point, including the existence of business flags of specified types (the business flag types you created earlier). See the embedded help for more details.
- **Service Investigation Order**: The field activity type and Field Task Type for field activities created by the investigation.
- **Completion Criteria**: The criteria used to determine if the service investigation order (field activity) should be completed and the result to be returned to the investigation. See the embedded help for more details.
- **Consumption Data Source**: Details that specify the types of consumption data will be displayed and how it is displayed in the **Meter Insights Investigation Viewer** portal, including the 360 Event Bar Profile and Final Values Overlay Profile extendable lookup values created earlier.
- **To Do Information**: The To Do Type and To Do Role used for To Do Entries created by the investigation. This should reference the To Do Type configured earlier.

Note: After you have created the appropriate service task types, return to the corresponding business flag type and select the correct service task type from the **Service Issue Monitor Type** drop-down list.

### Master Configuration

Create a Meter Insights Configuration master configuration to capture key details used by Oracle Utilities Analytics Insights when creating insights including:

- Tolerances details related to specific types of insights (Flipped Meter, Unauthorized Consumption, and Leak Detection)
- Core sources of consumption for electric, gas, and water service types (including service point types, and unit of measure/time of use/service quantity identified combinations for each)

The master configuration is synchronized with Oracle Utilities Analytics Insights, and is used by Oracle Utilities Analytics Insights when generating insights for potential investigation.

# Usage and Event Integration

Oracle Utilities Smart Grid Gateway adapters can be configured to publish initial measurement and device event data for use in external analytics applications or other external systems.


Copyright © 2026, Oracle and/or its affiliates.

Published data is posted to a user-defined directory that Oracle Utilities Analytics Insights can monitor for import.

Initial measurement and device event data is published in the “native” data format (the format of the initial measurement and device event seeder business objects). This format includes normalized unit of measure, condition, and device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

> **Note**
>
> Initial measurement data published via this feature is published prior to VEE processing. In addition, filtering can NOT be applied to initial measurement or device event data published via this feature.

- Configuring Analytics Usage and Event Integration

## Configuring Analytics Usage and Event Integration

Oracle Utilities Smart Grid Gateway adapters use OSB and BPEL components when publishing initial measurement and device event data for integration with external analytics applications.

### OSB Projects

The following “CM” Oracle Service Bus projects are used to configure the integration between SGG and external analytics applications:

| Adapter | OSB Projects |
|---|---|
| Landis+Gyr | SGG-D3-USAGE-CM (used to publish usage)<br>SGG-D3-EVENT-CM (used to publish device events) |
| NES | SGG-D4-USAGE-CM (used to publish usage)<br>SGG-D4-EVENT-CM (used to publish device events) |
| MV90 | SGG-D5-USAGE-CM (used to publish usage) |
| Sensus | SGG-D6-USAGE-CM (used to publish usage)<br>SGG-D6-EVENT-CM (used to publish device events) |
| Silver Spring Networks | SGG-D7-SSNXML-CM (used to publish usage)<br>SGG-D7-CSV-CM (used to publish device events) |
| Itron OpenWay | SGG-D8-ITRONXML-CM (used to publish usage)<br>SGG-D8-EXCEPTION-CM (used to publish device events) |
| Adapter Development Kit | SGG-DG-SEEDER-CM (used to publish usage and device events) |

The following components provided with these projects are used in publishing measurement data and device events to Oracle Utilities Analytics Insights:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This business service is specified in the EnvironmentSettings.xq file (see below).


Copyright © 2026, Oracle and/or its affiliates.

- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

### Enabling Data Publishing

Publishing initial measurement and device event data is enabled by referencing a publisher business service (DataRakerBusinessService) in the publishServices/service element in the EnvironmentSettings.xq files provided with the OSB projects listed above as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

### BPEL Composites

The SGGDRIntegration BPEL composite handles publishing the data to external analytics applications or other systems.

### Configuring Publishing Output

The following parameters can be used to configure details of how the data is provided to external analytics applications, including the directory where files are posted for external analytics applications to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

| Parameter | Description | Default Value |
|---|---|---|
| SGG_DR_INT_QUEUE | JNDI name of queue to publish SGG payloads.<br>This is the JMS queue defined in the **DataRakerBusinessService** business service. This should NOT be changed. | DataRakerQueue |
| SOA_DR_PUBLISH_SIZE | The number of records (SGG payloads) to accumulate in a published file. | 100 |
| SOA_DR_FILE_SIZE | The maximum file size for the accumulated (SGG payloads) file in kilobytes. | 524288 |
| SOA_DR_ELAPSED_TIME | The period of time in second which, when exceeded, causes a new outgoing file to be created. | 600 |
| SOA_DR_POLLING_FREQ | The polling frequency in seconds of the staging directory for new files. | 60 |
| SOA_DR_STAGING_DIR | Mount point/directory for the staging directory for accumulated SGG payload files.<br>This is used internally and should NOT be changed. | /spl/sploutput/staging |


Copyright © 2026, Oracle and/or its affiliates.

| Parameter | Description | Default Value |
|---|---|---|
| SOA_DR_INTEGRATION_DIR | Mount point/directory from which external analytics applications will consume the converted XML files. | /spl/sploutput/int |

# Oracle Utilities Analytics Visualization

- Timeliness and Quality Aggregations Overview
- Consumption Analytics Aggregations Overview

## Timeliness and Quality Aggregations Overview

Oracle Utilities Meter Solution Cloud Service and Oracle Utilities Customer Cloud Service can be integrated with Oracle Utilities Analytics Visualization (OUAV) to allow users to view analytics data based on the quality and timeliness of measurement data.

This section describes integration of Oracle Utilities Meter Solution Cloud Service and Oracle Utilities Customer Cloud Service with Oracle Utilities Analytics Visualization (OUAV).

See Operational Analytics in the *Administrative User Guide* for general information about integrating with Oracle Utilities Analytics Visualization.

- Base Package Timeliness and Quality Aggregation Measuring Components and Measurement Business Objects
- Configuring Timeliness and Quality Aggregation for Use with Oracle Utilities Analytics Visualization
- Timeliness and Quality Aggregation Processing for Use with Oracle Utilities Analytics Visualization

## Base Package Timeliness and Quality Aggregation Measuring Components and Measurement Business Objects

Oracle Utilities Meter Data Management and Oracle Utilities Customer To Meter include a number of standard timeliness and quality aggregation processes delivered as part of the base product that can be used with Oracle Utilities Analytics Visualization (OUAV). These aggregate measurement data (including quantities and counts) for constituent measuring components based on the following dimensions: Postal, City, Head-End, Device Type, Usage Calculation Group, Market and Service Provider, and Service Type.

The table below lists the names (and business object codes) for these aggregators, along with the corresponding Oracle Utilities Analytics Visualization subject area.


Copyright © 2026, Oracle and/or its affiliates.

| Aggregator | Description |
|---|---|
| Measurement Timeliness Quantity Aggregator<br>Business Object: D2-MsrmtTimelinessQuantityAggr<br>OUAV Subject Area: Meter - Aggregated Timeliness Quantity | Aggregates measurement quantities that arrived on time or are late. |
| Measurement Timeliness Count Aggregator<br>Business Object: D2-MsrmtTimelinessCountAggr<br>OUAV Subject Area: Meter - Aggregated Timeliness Count | Aggregates measurement counts that arrived on time or are late. |
| Measurement Quality Count Aggregator<br>Business Object: D2-MsrmtQualityCountAggregator<br>OUAV Subject Area: Meter - Aggregated Measurement Count | Aggregates measurement counts and distributes the constituents' counts across Value Identifiers based on condition codes. |
| Measurement Measured Quantity Aggregator<br>Business Object: D2-MeasuredQuantityAggregator<br>OUAV Subject Area: Meter - Aggregated Measurement Quantity | Aggregates measurement quantities and distributes the constituents' measurements across Value Identifiers based on condition codes. |

Refer to the Oracle Utilities Analytics Visualization User Guide for more information about Oracle Utilities Analytics Visualization subject areas.

In addition, the base package also includes measurement business objects used to store aggregated measurement data. These include:

- Measured Quantity Measurement (D2-MeasuredQuantityMsrmt)
- Quality Count Measurement (D2-QualityCountMsrmt)
- Timeliness Count Measurement (D2-TimelinessCountMsrmt)
- Timeliness Quantity Measurement (D2-TimelinessQuantityMsrmt)

## Configuring Timeliness and Quality Aggregation for Use with Oracle Utilities Analytics Visualization

This topic outlines the configuration needed for timeliness and quality aggregations used with Oracle Utilities Analytics Visualization.

### Master Configuration

Create a Timeliness Master Configuration, defining the following:

- **Heating Degree Days Factor**: A number-based factor that contains heating degree days. These are aggregated and stored on the Measurement Measured Quantity Aggregator.
- **Cooling Degree Days Factor**: A number-based factor that contains cooling degree days. These are aggregated and stored on the Measurement Measured Quantity Aggregator.
- **Number of Hours to Consider IMD as Late**: The number of hours after which an initial measurement is considered late.


Copyright © 2026, Oracle and/or its affiliates.

- **Buckets**: Value Identifiers that specify the late buckets used by the Measurement Timeliness Count and Measurement Timeliness Quantity aggregators.

#### Extendable Lookup: BI Aggregation — Usage Group Mapping

If your implementation makes use of dynamic usage calculation group determination on usage subscriptions (via the **Calculation Group Determination Override Algorithm** on the usage subscription type, the **Rate History** collection on the usage subscription, and the Customer Rate Schedule extendable lookup (D2-CCBRateScheduleLookup)), you will need to configure the BI Aggregation Usage Group Mapping extendable lookup (D2-UsageCalculationGroupLookup) with the usage calculation groups you wish to aggregate. If in the future you add new usage calculation groups, you will also need to add those usage calculation groups to this extendable lookup if you want them to be aggregated.

Note that an aggregation measuring component will be created for all dimension combinations and each usage calculation group in the extendable lookup (this is done to alleviate the need to execute the usage group determination algorithm for all usage subscriptions during the dimension scanning process).

> **Note**
>
> Configuring this extendable lookup is required ONLY if using dynamic usage calculation group determination.

#### Measuring Component Types

Create a master aggregator measuring component type and sub aggregator measuring component types for the four types of aggregators outlined above. Details for each aggregator measuring component type are listed below.

> **Note**
>
> When creating the master aggregator measuring component type, we recommend setting the **Aggregation Horizon** at 7 days or less. See Aggregation Recommendations for more information.

#### Measurement Measured Quantity Aggregator

- **Measuring Component Type Business Object**: Master Measurement Aggregator Type (D2-MasterMsrmtAggregatorType)
- **Measuring Component Business Object**: Measurement Measured Quantity Aggregator (D2-MeasuredQuantityAggregator)
- **Measurement Business Object**: Measured Quantity Measurement (D2-MeasuredQuantityMsrmt)
- **Value Identifiers** (recommended):

| Value Identifier Type | Description | Unit of Measure |
|---|---|---|
| Measurement | kWh | Kilowatt-Hours |
| Value 01 | Regular kWh | Kilowatt-Hours |
| Value 02 | Estimated kWh | Kilowatt-Hours |


Copyright © 2026, Oracle and/or its affiliates.

| Value Identifier Type | Description | Unit of Measure |
|---|---|---|
| Value 03 | User-Edited kWh | Kilowatt-Hours |
| Value 06 | Count | Count |
| Value 07 | Heating Degree | Fahrenheit |
| Value 08 | Cooling Degree | Fahrenheit |
| Value 09 | Ave kWh | Kilowatt-Hours |

- **Sub Aggregator MC Types** (see below):
  - Measurement Quality Count Aggregator Type
  - Measurement Timeliness Count Aggregator Type
  - Measurement Timeliness Quantity Aggregator Type

**Measurement Quality Count Aggregator**

- **Measuring Component Type Business Object**: Sub Measurement Aggregator Type (D2-SubMeasurementAggType)
- **Measuring Component Business Object**: Measurement Quality Count Aggregator (D2-MsrmtQualityCountAggregator)
- **Measurement Business Object**: Quality Count Measurement (D2-QualityCountMsrmt)
- **Value Identifiers** (recommended):

| Value Identifier Type | Description | Unit of Measure |
|---|---|---|
| Measurement | kWh | Kilowatt-Hours |
| Value 01 | Regular Count | Count |
| Value 02 | Estimated Count | Count |
| Value 03 | User-Edited Count | Count |
| Value 04 | N Meas/No IMD Count | Count |
| Value 05 | No Meas/IMD Exists Count | Count |
| Value 06 | Read Outage Count | Count |
| Value 07 | Read Other Count | Count |
| Value 08 | Missing Count | Count |

**Measurement Timeliness Count Aggregator**

- **Measuring Component Type Business Object**: Sub Measurement Aggregator Type (D2-SubMeasurementAggType)
- **Measuring Component Business Object**: Measurement Timeliness Count Aggregator (D2-MsrmtTimelinessCountAggr)
- **Measurement Business Object**: Timeliness Count Measurement (D2-TimelinessCountMsrmt)
- **Value Identifiers** (recommended):

| Value Identifier Type | Description | Unit of Measure |
|---|---|---|
| Measurement | kWh | Kilowatt-Hours |


Copyright © 2026, Oracle and/or its affiliates.

| Value Identifier Type | Description | Unit of Measure |
|---|---|---|
| Value 01 | On Time Count | Count |
| Value 02 | Late Count | Count |
| Value 03 | Very Late Count | Count |
| Value 04 | Very, Very Late Count | Count |
| Value 05 | Missing Count | Count |

#### Measurement Timeliness Quantity Aggregator

- **Measuring Component Type Business Object**: Sub Measurement Aggregator Type(D2-SubMeasurementAggType)
- **Measuring Component Business Object**: Measurement Timeliness Quantity Aggregator (D2-MsrmtTimelinessQuantityAggr)
- **Measurement Business Object**: Timeliness Quantity Measurement (D2-TimelinessQuantityMsrmt)
- **Value Identifiers** (recommended):

| Value Identifier Type | Description | Unit of Measure |
|---|---|---|
| Measurement | kWh | Kilowatt-Hours |
| Value 01 | On Time kWh | Kilowatt-Hours |
| Value 02 | Late kWh | Kilowatt-Hours |
| Value 03 | Very Late kWh | Kilowatt-Hours |
| Value 04 | Very, Very Late kWh | Kilowatt-Hours |

#### Activity Type

Create a Measurement Quantity Scanner activity type, based on the following business objects:

- **Business Object**: Measurement Quantity Scanner Type
- **Related Transaction Business Object**: Measurement Quantity Scanner

## Timeliness and Quality Aggregation Processing for Use with Oracle Utilities Analytics Visualization

This topic describes the processes involved in running timeliness and quality aggregation processing for use with Oracle Utilities Analytics Visualization.

#### Execute Measurement Quantity Scan

Create an activity of the Measurement Quantity Scanner activity type, and process the activity (by clicking **Scan** in the **Record Action** section in the **Activity** zone).

This process uses the Measurement Quality and Timeliness Dimension Scanner batch control (D1-MQTDS) which creates measuring component aggregators that will also serve as the measuring component dimensions. This process will create a measuring component aggregator for each unique combination of Postal Code, City, Device Type, Head-End System,


Copyright © 2026, Oracle and/or its affiliates.

Usage Group, Market, Service Type, Geographic Code, Manufacturer/Model, Service Provider (where Market Relationship Type is specified on the dimension scanner activity).

> **Note**
>
> The Measurement Quality and Timeliness Dimension Scanner batch control (D1-MQTDS) supports multi-threading.

This step will create aggregator measuring components for all **Valid MC Types to Aggregate** defined on the master aggregator type (Measurement Measured Quantity Aggregator).

The created aggregator measuring component's characteristics will contain the Postal Code, City, Device Type, Head-End System, Usage Group, Market, Service Type, Geographic Code, Manufacturer/Model, Service Provider information.

**Execute Aggregation**

Once the measuring component aggregators have been created, run the Aggregation Monitor (D2-AGG) batch process. This will process ALL active master aggregators (together with the linked sub aggregators) that satisfy the batch filter parameters. Aggregated measurements will be created for the span of the aggregator horizon which is the current date less the horizon days in intervals defined on the master aggregation measuring component type (Measurement Measured Quantity Aggregator).

Condition Top Range and Condition Bottom Range parameters on the Aggregate Measurement Counts and Quantity (D2-AGG-MCQ) algorithm (an Enter algorithm on the Aggregate state of the Measurement Measured Quantity Aggregator (D2-MeasuredQuantityAggregator) business object) define condition ranges for quantity and count aggregation buckets.

This batch process should be run on a scheduled reoccurring basis.

**Aggregated Data**

Aggregated measurements are created in the Measurement (D1_MSRMT) and Aggregation Measurement (D1_AGG_MSRMT) tables, using the following base package measurement business objects:

- Measured Quantity Measurement (D2-MeasuredQuantityMsrmt)
- Quality Count Measurement (D2-QualityCountMsrmt)
- Timeliness Count Measurement (D2-TimelinessCountMsrmt)
- Timeliness Quantity Measurement (D2-TimelinessQuantityMsrmt)

Oracle Utilities Analytics Visualization Fact views access the Aggregation Measurement table.

Oracle Utilities Analytics Visualization Dimension views access the Measuring Component and Measuring Component Characteristics tables.

# Consumption Analytics Aggregations Overview

This section outlines the configuration needed to support integration with Oracle Utilities Analytics Visualization for the following consumption-based analytics:

- Usage by Network Location
- Usage by Region


Copyright © 2026, Oracle and/or its affiliates.

- Usage by Customer Classification
- Unaccounted for Losses

These analytics are accessed in Oracle Utilities Analytics Visualization (OUAV) via the following subject areas:

| Analytics | OUAV Subject Areas |
|---|---|
| Usage by Network Location<br>Usage by Region | Meter - Distribution Nodes |
| Usage by Customer Classification | Meter - Customer Classification |
| Unaccounted for Losses | Meter - Downstream Usage<br>Meter - Downstream Service Points<br>Meter - Facility Usage<br>Meter - Facilities |

- Configuring Dynamic Aggregation for Consumption Analytics
- Dynamic Aggregation Processing for Consumption Analytics

## Configuring Dynamic Aggregation for Consumption Analytics

This section outlines the configuration needed for consumption analytics aggregations used with Oracle Utilities Analytics Visualization.

Consumption analytics leverage data generated through dynamic aggregation, and requires configuration of the following objects:

- Data Sources
- Aggregator Measuring Component Types
- Aggregation Groups
- Measuring Component Sets

Refer to **Dynamic Aggregation** for more information about configuring dynamic aggregation.

Note: The configuration outlined below is based on data in the demonstration database. This sample configuration uses Electric service for Usage by Network Location, Region, and Customer Classification, and Gas service for Unaccounted for Losses.

### Data Sources

Create the following data sources, based on the Measurements from US Service Points (D1-DataSourceMeasurements) business object:

**For Usage by Network Location, Usage by Region, and Usage by Customer Classification (Electric)**

| Field / Section | Value |
|---|---|
| Data Source | CM-E-KWH-60-TRANS |
| Description | Electric kWh 60min Transformer Interval Measurements |
| Status | Active |
| Data Source Class | Measurements from US Service Points |
| Interval / Scalar | Interval |


Copyright © 2026, Oracle and/or its affiliates.

| Field / Section | Value |
|---|---|
| Sourced Measuring Component Type | |

| Measuring Component Type | Value Identifier Type |
|---|---|
| Measuring Component Types for measuring components to be aggregated | Measurement |

**For Unaccounted for Losses (Gas)**

| Field / Section | Value |
|---|---|
| Data Source | CM-G-CCF-60-FACIL |
| Description | Gas CCF 60min Facility Interval Measurements |
| Status | Active |
| Data Source Class | Measurements from US Service Points |
| Interval / Scalar | Interval |
| Sourced Measuring Component Type | |

| Measuring Component Type | Value Identifier Type |
|---|---|
| Measuring Component Types for measuring components to be aggregated | Measurement |

### Aggregator Measuring Component Types

Create the following aggregator measuring component types, based on the Dynamic Aggregator - Type (D1-AggregationIntervalType) business object:

**For Usage by Network Location, Usage by Region, and Usage by Customer Classification (Electric)**

| Field / Section | Value |
|---|---|
| Measuring Component Type | CM-E-I-AGG-KWH-60-TRANS |
| Description | Electric - Interval - Aggregation - kWh 60min Transformer Load |
| Measuring Component Business Object | Dynamic Aggregator - Interval (D1-AggregatorInterval) |
| Measurement Business Object | Aggregation Measurement (D1-AggregationMeasurement) |
| Measuring Component Class | Aggregator |
| Service Type | Electric |
| Allow Negative Consumption | Allowed |
| Consumptive/Subtractive | Consumptive |
| Interval Size | 01:00:00 |
| Data Source | Electric kWh 60min Transformer Interval Measurements |


Copyright © 2026, Oracle and/or its affiliates.

| Field / Section | Value |
|---|---|
| Value Identifiers | |

| Value Identifier Type | Short Hand Description | Unit of Measure |
|---|---|---|
| Measurement | kWh | Kilowatt-Hours |

**For Unaccounted for Losses (Gas)**

| Field / Section | Value |
|---|---|
| Measuring Component Type | CM-G-I-AGG-CCF-60-FACIL |
| Description | Gas - Interval - Aggregation - CCF 60min Downstream |
| Measuring Component Business Object | Dynamic Aggregator - Interval (D1-AggregatorInterval) |
| Measurement Business Object | Aggregation Measurement (D1-AggregationMeasurement) |
| Measuring Component Class | Aggregator |
| Service Type | Gas |
| Allow Negative Consumption | Allowed |
| Consumptive/Subtractive | Consumptive |
| Interval Size | 01:00:00 |
| Data Source | Gas CCF 60min Facility Interval Measurements |
| Value Identifiers | |

| Value Identifier Type | Short Hand Description | Unit of Measure | SQI | Value Calculation Method |
|---|---|---|---|---|
| Measurement | CCF | 100 Cubic Feet (Gas) | | Sum |
| Value 01 | Count | Count | | Count |
| Value 02 | Average | 100 Cubic Feet (Gas) | Average | Average |
| Value 03 | Min | 100 Cubic Feet (Gas) | Minimum | Min |
| Value 04 | Max | 100 Cubic Feet (Gas) | Maximum | Max |

### Aggregation Groups

Create the following aggregation groups based on the Aggregation Group (D1-AggregationGroup) business object:

**For Usage by Network Location and Usage by Region (Electric)**

| Field / Section | Value |
|---|---|
| Aggregation Group | CM-E-CONS-TRANS |
| Description | Electric Dynamic Consumption |


Copyright © 2026, Oracle and/or its affiliates.

| Field / Section | Value |
|---|---|
| Status | Finalized |
| Aggregation Category | Analytics |
| Time Zone | US Pacific Time |
| Aggregation Schedule | • **Aggregation Schedule Type**: Defined Lag<br>• **Aggregation Cut-Off Time**: 0:00:00<br>• **Aggregation Horizon**: 24:00:00<br>• **Aggregation Lag**: 48:00:00 |
| Aggregation Measuring Component Sets | |

| Sequence | Measuring Component Set | Initiation Method | Measuring Component Set Class |
|---|---|---|---|
| 10 | Electric - Consumption - Transformer - Interval | Manual | Aggregation Foundation |

### For Usage by Customer Classification (Electric)

| Field / Section | Value |
|---|---|
| Aggregation Group | CM-E-CONS-PROF |
| Description | Consumption - Profile Class |
| Status | Finalized |
| Aggregation Category | Analytics |
| Time Zone | |
| Aggregation Schedule | • **Aggregation Schedule Type**: Defined Lag<br>• **Aggregation Cut-Off Time**: 0:00:00<br>• **Aggregation Horizon**: 24:00:00<br>• **Aggregation Lag**: 48:00:00 |
| Aggregation Measuring Component Sets | |

| Sequence | Measuring Component Set | Initiation Method | Measuring Component Set Class |
|---|---|---|---|
| 10 | Consumption - Profile Class - Interval | Manual | Aggregation Foundation |

### For Unaccounted for Losses (Gas)

| Field / Section | Value |
|---|---|
| Aggregation Group | CM-G-LOSS |
| Description | Gas Dynamic Consumption |
| Status | Finalized |
| Aggregation Category | Analytics |
| Time Zone | US Pacific Time |


Copyright © 2026, Oracle and/or its affiliates.

| Field / Section | Value |
|---|---|
| Aggregation Schedule | • **Aggregation Schedule Type**: Defined Lag<br>• **Aggregation Cut-Off Time**: 0:00:00<br>• **Aggregation Horizon**: 24:00:00<br>• **Aggregation Lag**: 48:00:00 |
| Aggregation Measuring Component Sets | |

| Sequence | Measuring Component Set | Initiation Method | Measuring Component Set Class |
|---|---|---|---|
| 10 | Gas - Consumption - Downstream - Interval | Automatic | Aggregation Foundation |

### Measuring Component Sets

Create the following measuring component sets based on the Aggregation Foundation Set (D1-AggregationFoundationSet) business object:

**For Usage by Network Location and Usage by Region (Electric)**

| Field / Section | Value |
|---|---|
| Measuring Component Set | CM-E-CONS-TRANS |
| Status | Active |
| Description | Electric - Consumption - Transformer - Interval |
| Measuring Component Type | Electric - Interval - Aggregation - kWh 60min Transformer Load |
| Aggregation Group | Electric Dynamic Consumption |
| Measuring Component Set Class | Aggregation Foundation |
| Allow Partial Constituent Aggregation | Yes |
| Extract | Do Not Extract |
| General Processing | |
| Criteria Processing | • **Non-Individually Managed Formula**: TFORM |
| Dimensions | |

| Source Type | Source Entity | Column/Type | Represent Missing Data |
|---|---|---|---|
| Table | Service Point - MDM | City | No |
| Table | Service Point - MDM | Postal | No |
| Table (Demo) | Service Point Facility - MDM | Facility ID | No |
| Table (Demo) | Service Point Facility - MDM | Facility Relationship Type | No |


Copyright © 2026, Oracle and/or its affiliates.

| Field / Section | Value |
|---|---|
| Criteria | • **Criteria Name**: TFORM<br>• **Source Type**: Table (Demo)<br>• **Source Entity**: Service Point Facility - MDM<br>• **Column/Type**: Facility Relationship Type<br>• **Criteria Type**: Include<br>• **Value**: Commodity Distribution |

### For Usage by Customer Classification (Electric)

| Field / Section | Value |
|---|---|
| Measuring Component Set | CM-E-CONS-PROF |
| Status | Active |
| Description | Consumption - Profile Class - Interval |
| Measuring Component Type | Electric - Interval - Aggregation - kWh 60min Transformer Load |
| Aggregation Group | Consumption - Profile Class |
| Measuring Component Set Class | Aggregation Foundation |
| Allow Partial Constituent Aggregation | Yes |
| Extract | Do Not Extract |
| General Processing | • **Sequence**: 10<br>• **Initiation Method**: Manual<br>• **Imported from External Source**: No |
| Criteria Processing | |
| Dimensions | |

| Source Type | Source Entity | Column/Type | Represent Missing Data |
|---|---|---|---|
| Characteristic * | Usage Subscription Characteristics* | Procurement Class* | No* |
| Table | Service Point - MDM | City | No |
| Table | Service Point - MDM | Postal | No |
| Table | Usage Subscription | Usage Subscription Type | No |
| Table (Demo) | Service Point Facility - MDM | Facility ID | No |

* Optional

| | |
|---|---|
| Criteria | |

### For Unaccounted for Losses (Gas)

| Field / Section | Value |
|---|---|
| Measuring Component Set | CM-G-LOSS |


Copyright © 2026, Oracle and/or its affiliates.

<table>
<tr><th>Field / Section</th><th>Value</th></tr>
<tr><td>Status</td><td>Active</td></tr>
<tr><td>Description</td><td>Gas - Consumption - Downstream - Interval</td></tr>
<tr><td>Measuring Component Type</td><td>Gas - Interval - Aggregation - CCF 60min Downstream</td></tr>
<tr><td>Aggregation Group</td><td>Gas Dynamic Consumption</td></tr>
<tr><td>Measuring Component Set Class</td><td>Aggregation Foundation</td></tr>
<tr><td>Allow Partial Constituent Aggregation</td><td>Yes</td></tr>
<tr><td>Extract</td><td>Do Not Extract</td></tr>
<tr><td>General Processing</td><td></td></tr>
<tr><td>Criteria Processing</td><td>Non-Individually Managed Formula: TFORM</td></tr>
<tr><td>Dimensions</td><td>
<table>
<tr><th>Source Type</th><th>Source Entity</th><th>Column/Type</th><th>Represent Missing Data</th></tr>
<tr><td>Table</td><td>Service Point - MDM</td><td>City</td><td>No</td></tr>
<tr><td>Table</td><td>Service Point - MDM</td><td>Postal</td><td>No</td></tr>
<tr><td>Table (Demo)</td><td>Service Point Facility - MDM</td><td>Facility ID</td><td>No</td></tr>
<tr><td>Table (Demo)</td><td>Service Point Facility - MDM</td><td>Facility Relationship Type</td><td>No</td></tr>
</table>
</td></tr>
<tr><td>Criteria</td><td>
<ul>
<li>Criteria Name: TFORM</li>
<li>Source Type: Table (Demo)</li>
<li>Source Entity: Service Point Facility - MDM</li>
<li>Column/Type: Facility Relationship Type</li>
<li>Criteria Type: Include</li>
<li>Value: Commodity Distribution</li>
</ul>
</td></tr>
</table>

## Dynamic Aggregation Processing for Consumption Analytics

Processing of dynamic aggregation for consumption analytics uses the standard approach to dynamic aggregation execution. Refer to **Dynamic Aggregation** in the *Business User Guide* for more information about dynamic aggregation execution and processing.

# Oracle Utilities Network Management System


- Overview
- Configuring Device Event Notification Suppression


## Overview

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with Oracle Utilities Network Management System. In an integration between Oracle Utilities Meter Data Management and Oracle Utilities Network Management System:



Copyright © 2026, Oracle and/or its affiliates.

- Oracle Utilities Smart Grid Gateway can notify Oracle Utilities Network Management Systems of device events that have been received (e.g. last gasp and power up).
- Oracle Utilities Network Management Systems can request:
  - Oracle Utilities Smart Grid Gateway to perform various smart meter commands for a device or list of devices (e.g. device status check)
  - Oracle Utilities Smart Grid Gateway to suppress device event notifications for a certain set of device event types.

## Configuring Device Event Notification Suppression

Oracle Utilities Meter Data Management includes the following components used when integrating with Oracle Utilities Network Management Systems:

### Service Provider Processing Methods

The following processing methods should be associated to the service provider configured for Oracle Utilities Network Management System:

| Processing Role | Description |
|---|---|
| Bulk Response | This role will identify the appropriate message to send with the results of a bulk command execution. For example, this would be used if a large list of devices had been supplied to execute device status check in order to verify outage status. |
| Response - Success | This role will identify the appropriate message to send for a successful smart meter command |
| Send Device Event | This role sets up the subscription for device events that Oracle Utilities Network Management System has interest in. |
| Suppression Notification | This role identifies the appropriate message to send when a notification has been set or removed. |

### Activity Types

The following processing activity types should be configured:

| Activity Type | Description |
|---|---|
| Bulk Request Header | Configuration for the bulk request header activity type which is used in making smart meter command requests for a list of meters. |
| Bulk Response | Configuration for the bulk response activity type which organizes the response for bulk requests. |
| Device Event Notification Suppression | Identifies the device event categories and types that are subject to suppression. |

# Oracle Utilities Customer Self Service


Copyright © 2026, Oracle and/or its affiliates.

- Overview
- Configuring Rate Compare Usage Adjustment Profiles


# Overview

Oracle Utilities Meter Data Management can be integrated with Oracle Utilities Customer Self Service (OUCSS) to allow utility customers to:

- View their usage data
- Create self-service meter readings (i.e. submit their own meter readings)
- Perform rate comparisons (via integration between CC&B and MDM)

### Master Configuration

The following master configurations are used to configure the sync process between MDM and OUCSS:

| Master Configuration | Description |
|---|---|
| Self Service Master Configuration | This master configuration is used for general configuration related to the MDM to OUCSS interface. This includes a variety of configuration including: temperature source, rate comparison, supported usage calculation groups, etc. |

### Supporting Oracle Utilities Meter Data Management Services

Oracle Utilities Meter Data Management provides the following services used by Oracle Utilities Customer Self Service:

| Service Name | Description | Logic | Inbound Web Service |
|---|---|---|---|
| **Create Self-Service Meter Read** | Allows users to submit their own meter reads via the Customer Self Service application. | Service Script: WX-CrSSMRead | WX-CreateSelfServiceMeterRead |
| **Get Scalar Consumption Summary** | Retrieves consumption data for display in the Customer Self Service application. | Service Script: WX-GetSCsum | WX-GetScalarConsumptionSummary |
| **Get Usage Overview** | Retrieves an overview of a customer's usage for a user-specified duration for display in the Customer Self Service application | Service Script: WX-GetUsgOVw | WX-GetUsageOverview |
| **Multiple Account TOU Usages by Service Request Type** | Uses the Get Usage Details service to retrieve usage for a list of accounts for display in the Customer Self Service application. Each account's usage is summarized by service type, UOM and SQI. | Service Script: WX-MulAccTOU | WX-MultipleAccountTOUUsagesByServiceType |



Copyright © 2026, Oracle and/or its affiliates.

| | | | |
|---|---|---|---|
| **Multiple Account Usages by Service Request Type** | Uses the Get Usage Details service to retrieve usage for a list of accounts for display in the Customer Self Service application. Each account's usage is summarized by service type, UOM and SQI. | Service Script:<br>WX-MulAccCmp | WX-MultipleAccountUsagesByServiceType |
| **Multiple Account Usage Download Request** | Uses the Get Usage Details service to retrieve usage for a list of accounts (by usage subscription) for display in the Customer Self Service application. | Service Script:<br>WX-MulAccUsg | WX-MultipleAccountUsagesDownload |
| **Get Usage Details** | Retrieves usage details for a customer for a user-specific time period (year, month, day) for display in the Customer Self Service application | Business Service:<br>WX-RETWSSTOUMapping | WX-RETWSSTOUMappingService |
| **Usage Adjustment Retrieval** | Retrieves usage adjustment types based on a user-specified rate schedule. Used by the Rate Compare feature of the Customer Self Service application. | Service Script:<br>WX-RetUsgAdj | WX-RetUsgAdj<br>OR<br>WX-UsageAdjustmentRetrieval |

These services are based on the service scripts and business services noted above, and are invoked via the corresponding Inbound Web Services.

Refer to the *Oracle Utilities Customer Self Service Implementation Guide* for more information about integrating Oracle Utilities Meter Data Management with Oracle Utilities Customer Self Service.

# Configuring Rate Compare Usage Adjustment Profiles

This section describes the steps involved in setting up usage adjustment profiles in Oracle Utilities Meter Data Management for use with the Rate Compare feature in Oracle Utilities Customer Self- Service.

This section provides an overview of the steps involved in setting up usage adjustment profiles. The sections that follow provide additional details and examples for each type of data that must be configured.

1. Create a characteristic type and values for each type of usage adjustment to be made available for rate comparison purposes. For example, if you wanted to allow adjustments for installation of solar panels, use of an energy-efficient appliance, or use of an electric vehicle, you would create three characteristic types (one for each type of adjustment), and values for each option within each type (each type of appliance or electric vehicle supported).


Copyright © 2026, Oracle and/or its affiliates.

2. Create profile (stand-alone) measuring components that correspond to each characteristic type value.
3. Create profile data for each measuring component. This is interval data that represents the impact of the usage adjustment on a customer's consumption.
4. Create a profile factor for each usage adjustment type.
5. Create factor values for each factor. These values correspond to the characteristic type value, and link characteristic type values to profile measuring components.
6. Create one (or more) Profile Accumulation usage calculation rules that will apply the usage adjustment profile to the customer's interval consumption when calculating usage for the rate comparison request.
7. Create entries in the Customer Rate Schedule extendable lookup to associate the usage calculation group that contains the "Profile Accumulation" usage calculation rule to a rate schedule in Oracle Utilities Customer Care and Billing.
8. Create entries in the Usage Adjustment Types extendable lookup for each usage adjustment type.
9. Set up the Rate Compare Configuration section of the Self-Service Master Configuration to link usage adjustment factors and usage adjustment types to rate schedules Oracle Utilities Customer Care and Billing (defined in the Customer Rate Schedule extendable lookup).

### Characteristic Types

Create a characteristic type and values for each type of usage adjustment you wish to make available to customer self-service users. For example, to create a usage adjustment profile for use of an electric vehicle, you would create an "electric vehicle" characteristic type and define values for each type of electric vehicle users can select.

Example Characteristic Type:

- Characteristic Type: ELEC_VEH
- Description: Electric Vehicles
- Type of Char Value: Predefined Value
- Characteristic Values:

| Characteristic Value | Description |
|---|---|
| LEAF | Nissan Leaf |
| TESLA | Tesla Model S |
| VOLT | Chevrolet Volt |

### Profile Measuring Components

Create profile measuring components for each characteristic type value. These measuring components will be used to store profile data for each type of usage adjustment.

**Note**: The base package does not include stand-alone measuring component/measuring component type business objects, but the demonstration database contains "Standalone Interval" and "Standalone Interval Measuring Component Type" business objects that can be used to create profile measuring components and types.

Example Profile Measuring Component Type:

- Measuring Component Type: KWH-PROFILE


Copyright © 2026, Oracle and/or its affiliates.

- Description: KWH Profile
- Measuring Component Business Object: Standalone Interval (demo)
- Measurement Business Object: Measurement
- Service Type: Electric
- Allow Negative Consumption: Allowed
- Consumptive / Subtractive: Consumptive
- Seconds Per Interval: 01:00:00
- Value Identifiers:
- Value Identifier Type: Measurement
- Short-Hand Description: kWh
- UOM: Kilowatt-Hours

Example "Nissan Leaf" Profile Measuring Component:

- Measuring Component Type: KWH Profile
- Number of Digits Left: 5
- Number of Digits Right: 5
- Time Zone: US Pacific Time
- Status: Active
- How To Use: Additive
- External ID: Nissan Leaf

Example "Tesla Model S" Profile Measuring Component:

- Measuring Component Type: KWH Profile
- Number of Digits Left: 5
- Number of Digits Right: 5
- Time Zone: US Pacific Time
- Status: Active
- How To Use: Additive
- External ID: Tesla Model S

Example "Chevrolet Volt" Profile Measuring Component:

- Measuring Component Type: KWH Profile
- Number of Digits Left: 5
- Number of Digits Right: 5
- Time Zone: US Pacific Time
- Status: Active
- How To Use: Additive
- External ID: Chevrolet Volt


Copyright © 2026, Oracle and/or its affiliates.

### Profile Data

Create profile data for each profile measuring component. This data represents the impact of the usage adjustment on a customer's consumption.

Note that this profile data can (and often will) include negative interval values to represent the difference in consumption applicable for the usage adjustment type. For example, if an energy-efficient electric clothes dryer uses an average of 30 kilowatt hours less per month than an average electric clothes dryer, profile data for that appliance might be a "straight line" hourly profile (a profile in which all intervals are of the same value) in which each value equals "0.042" (30 kWh per month divided by an average of 720 hours per month).

### Factors / Factor Values

Create a profile factor for each usage adjustment type. This factor should use the "Factor Characteristic Source N/A Algorithm" to derive the appropriate characteristic type value based on the factor value, and should reference the characteristic type created earlier.

Example Electric Vehicle Factor:

- **Factor**: ELECTRIC_VEHICLE
- **Description**: Electric Vehicle
- **Factor Class**: Profile
- **Characteristic Source Algorithm**: Factor Characteristic Source N/A Algorithm
- **Factor Characteristic Type**: Electric Vehicle

Example "Nissan Leaf" Factor Value

- **Factor: Electric Vehicle**
- **Factor Characteristic Type: Electric Vehicle**
- **Factor Characteristic Value: LEAF**
- **Effective Date/Time: 01-01-2014 12:00:00AM**
- **Profile: Nissan Leaf, KWH Profile**

Example "Tesla Model S" Factor Value

- **Factor: Electric Vehicle**
- **Factor Characteristic Type: Electric Vehicle**
- **Factor Characteristic Value: LEAF**
- **Effective Date/Time: 01-01-2014 12:00:00AM**
- **Profile: Tesla Model S, KWH - 60 Minutes**

Example "Chevrolet Volt" Factor Value

- **Factor: Electric Vehicle**
- **Factor Characteristic Type: Electric Vehicle**
- **Factor Characteristic Value: LEAF**
- **Effective Date/Time: 01-01-2014 12:00:00AM**
- **Profile: Chevrolet Volt / KWH - 60 Minutes**


Copyright © 2026, Oracle and/or its affiliates.

### Profile Accumulation Usage Calculation Group and Rule

Create a usage calculation group that contains a "Profile Accumulation" usage calculation rule. This rule will calculate usage by accumulating historical usage with profile data. based on a selected profile factor value.

**Note**: Profile Accumulation rules should use eligibility criteria to ensure they are only executed when the "Calculation Mode" on the usage transaction is set to "Hypothetical Calculation" (D2HC).

**Example**: Profile Accumulation Usage Calculation Rule

- **Usage Calculation Group: Electric Residential Interval KWH**
- **Usage Calculation Rule: KWH_PROFILE_ACCUMULATION**
- **Sequence: 10**
- **Description: KWH Profile Accumulation**
- **Category: Usage Calculation**
- **Vector Source Configuration:**
- **Vector Type**: Channels Linked to Usage Subscription
- **Unit of Measure**: Kilowatt-Hours
- **Time of Use**:
- **Service Quantity Identifier**:
- **Target SPI**: 01:00:00
- **Result Processing Configuration**:
- **Apply TOU Map to Derived Vector: Yes**
- **TOU Map: Summer / Winter, 15 minute interval**
- **Result Storage Configuration**:
- **Insert Primary SQ Entry:** Yes
- **Save Derived Vector:** No
- **Service Quantity Identifier:**
- **Extract Interval Data:** No

### Customer Rate Schedule Extendable Lookup

Create an entry in the "Customer Rate Schedule" extendable lookup to associate the usage calculation group that contains the "Profile Accumulation" usage calculation rule to an applicable rate schedule in Oracle Utilities Customer Care and Billing.

Example Customer Rate Schedule extendable lookup:

- **Rate**: E-INT-RES
- **Description**: Electric Residential Interval Rate
- **Default Usage Calculation Group**: Electric Residential Interval KWH (E-INT-RES)


Copyright © 2026, Oracle and/or its affiliates.

#### Usage Adjustment Types Extendable Lookup

Create an entry in the "Usage Adjustment Type" extendable lookup for each type of usage adjustment that will be available to customer self-service users. These entries are used in the Rate Compare Configuration section of the Self-Service Master Configuration (see below).

Example Electric Vehicle entry:

- **Usage Adjustment Type: ELEC_VEHICLE**
- **Description: Purchase of Electric Vehicle**
- **Override Description: Purchase of Electric Vehicle**
- **External Reference ID: Purchase of Electric Vehicle**

#### Self-Service Master Configuration - Rate Compare Configuration

Configure the "Rate Compare Configuration" section of the Self-Service Master Configuration to associate usage adjustment factors and usage adjustment types with an applicable rate schedule in Oracle Utilities Customer Care and Billing (defined in the Customer Rate Schedule extendable lookup).

Example Self-Service Master Configuration:

- **Factor Characteristic Type Indicating No Value Variation: N/A**
- **External Reference Factor Value Characteristic Type: External Reference ID**
- **Minimum Days of Usage Adjustment Data: 2**
- **Rate / Usage Adjustments:**
- **Rate**: Electric Residential - Interval KWH
- **Usage Adjustments**:
- Usage Adjustment Factor: Electric Vehicle
- Usage Adjustment Type: Purchase of Electric Vehicle

# Oracle Utilities Dataconnect / Opower


- Overview
- Consumption Extract
- Master Data Extract
- Extract Flat File Formats


## Overview

Oracle Utilities DataConnect facilitates extraction of data from Oracle Utilities Meter Data Management for use in external applications such as analytics applications and energy management systems. This section describes how Oracle Utilities DataConnect works and how to implement and configure the system to support data extract processing.

Oracle Utilities DataConnect can be used to extract two types of data:

- Measurement data for service points based on the Service Point Types defined for a Consumption Extract Type. When extracting usage measurement data, Oracle Utilities



Copyright © 2026, Oracle and/or its affiliates.

DataConnect extracts both interval measurement data as well as frequently read scalar usage data (which is converted into interval measurements during the extraction process).

- Individual service points and install events (based on Service Point Types defined in an extract algorithm).

The extraction processes used for each type of data are separate, but extracted data can be correlated in external systems through service point data included in each type of extract. All data extracts contain the following data elements that can be used for this correlation:

- Service Point ID: The service point ID in Oracle Utilities Meter Data Management
- CIS Service Point ID: The ID used for the service in a customer information system (CIS) such as Oracle Utilities Customer Care and Billing

External systems receiving data extracted from Oracle Utilities DataConnect can use either (or both) of these IDs to associate extract measurement data to extracted master data.

> **Note**
>
> There are two versions of DataConnect. The initial version is referred to as “Legacy DataConnect”. The current and recommended version is simply referred to as “DataConnect” or “Current DataConnect”. Implementations are recommended to use the current version. “Legacy DataConnect” remains available for implementations that used the initial version, but is not recommended for new implementations.

# Consumption Extract

When sending interval measurement usage data to an external system, both historical and current data needs to be extracted. Historical data can be extracted as part of an initial load process, and only needs to be provided during initial setup of the integration. Historical data should include data history for all active service points for a specified historical period. Current data should be extracted on a regular ongoing (or incremental) basis. However, in addition to sending current data, any historical corrections received by the system should be extracted as well.

### Master Configuration

When using Direct Measurement Processing, the **Use Measurement Extract Request for Historical Extracts** parameter on the MDM Master Configuration must be set to "Yes" in order to run historical extracts. When this is set to "Yes", change data capture information is recorded in the Measurement Extract Request table instead of the General Process table.

### Extract Requests

There are several types of consumption extract requests:

- **Initial Load**: Initial load extract requests are created and submitted manually via the Consumption Extract Request portal. Consumption extract requests are based on a specified Consumption Extract Type (see below) and extraction date range. An initial load request must be created and submitted for each consumption extract type defined in the system.
- **Incremental / Ongoing (Current Data)**: Incremental / ongoing extract requests can be manually created, but more often will be created via a batch process. The "Create Daily Consumption Extract Requests" batch control scans active consumption extract types and


Copyright © 2026, Oracle and/or its affiliates.

creates a request for each one that has Frequency of "Automated Daily." Ad-hoc incremental requests can be created and submitted manually if needed.

- **Historical Correction**: Historical correction extracts are created via batch process. Algorithms on the Finalized state of the initial measurement and measurement business objects determine if a finalized initial measurement or rederived values are historical corrections. These algorithms create records which are evaluated by a batch process which extracts the measurements for the related initial measurements.

### Consumption Extract Type

Consumption Extract Types define the specific parameters used when processing a consumption extract request. Consumption Extract Types control the service point type, type of measurement, and how the measurements are grouped into TOU periods if applicable. Consumption extract types also define the algorithm and batch processes to use when extracting data for different types of requests (initial load, incremental, and historical).

There are two consumption extract type business objects provided with the base package:

- **Energy Management Extract** (D2-ConsumptionExtractType): This business object retrieves interval data and converts it to a specified Target UOM and Interval Size. This business object does not support TOU mapping.
- **Energy Management Extract with TOU Mapping** (D2-ConsumptionExtractTypeTOU): This business object retrieves interval data, converts it to a specified Target UOM and Interval Size, and maps it to a specific TOU map prior to extraction.

### Historical Versus Current Data

The "Extract Through Date/Time" field on the Consumption Extract Type is used to differentiate between current data (the most recently extracted data) and historical corrections, and is set to the last date on which data was extracted for that extract type. For example, if data is extracted on June 1, 2015, the "Extract Through Date/Time" would be set to "June 1, 2015 12:00AM." If/when data is extracted the next day, "Extract Through Date/Time" would be updated to "June 2, 2015 12:00AM."

When evaluating data for extract:

- Interval data is considered current if its measurement date time is after the "Extract Through Date/Time".
- Interval data is considered a historical correction if its measurement date time is on or before the "Extract Through Date/Time".

Historical data changes to an initial measurement can be detected when it enters the Finalized state. If the initial measurement is determined to be for a historical period by comparing its end date/time against the "Extracted Through Date/Time" on the Consumption Extract Type, a record will be written in the General Process or Measurement Extract Request table for the initial measurement so that measurements for it can be extracted. In addition, re-derived values on final measurements can also trigger the creation of a record in the General Process or Measurement Extract Request table for related initial measurements.

The following algorithms are used in this process:

- The "Create General Process Record if IMD is Historical Correction" algorithm is used to determine if a finalized initial measurement data is a historical correction. If it is, the algorithm creates record in the General Process or Measurement Extract Request table for the initial measurement. This algorithm is provided in the base package, but not specified on initial measurement business objects by default. This algorithm should be defined as an Enter algorithm on the Finalized state of the initial measurement business objects.


Copyright © 2026, Oracle and/or its affiliates.

- The "Create General Process Record for Re-derived Values" algorithm creates records in the General Process or Measurement Extract Request table for initial measurements associated with re-derived values. Processing will proceed as if a historical correction came in through an initial measurement. This algorithm is provided in the base package, but not specified on measurement business object by default. This algorithm should be defined as an Enter algorithm on the Re-derive state of the final measurement business object.

### Consumption Extract Requests

Initial load and ongoing consumption extracts are created via Consumption Extract Requests. While extracts of these types can be created via adhoc submission of a batch job, requests are the preferred method for these types of consumption extracts.

The consumption extract request business object lifecycle includes logic that maintains and updates the "Extraction Through Date/Time" field on Consumption Extract Types, which is used to determine if daily requests should be created by the "Create Daily Consumption Extract Requests" batch control, and detect historical corrections.

A single business object is provided in the base package for consumption extract requests:

- **Consumption Request** (D2-IntervalDataExtRepository): This business object contains the information and lifecycle responsible for submitting the extract job, monitoring the run until it's finished, and then updating the Consumption Extract Type's "Extract Through Date/Time" on the Consumption Extract Type. This business object is based on the Request (F1-REQ) maintenance object.

The consumption extract process uses a set of base package algorithms to extract and format the interval data for export. These algorithms are specified in the '"Algorithms" section on the Consumption Extract Type as appropriate. These algorithms can be configured to allow for extraction of data for frequently-read scalar measuring components as well as interval measuring components. Frequently-read scalar measuring components are defined as scalar measuring components whose Read Method is set to "Automatic Read." When extracting measurements for frequently-read scalar measuring components, scalar measurements are converted to interval measurements as part of the extraction process. This conversion uses the profile associated with the measuring component type. If no profile can be found, the interval data uses a flat line profile.

**Initial Load / Incremental / Ongoing Requests**: the following algorithms are used to extract and format interval data for initial load and incremental / ongoing requests:

- **Extract Initial Load/Ongoing Consumption for DataConnect** (D2-IDEXTPRD): This algorithm retrieves a service point's consumption for a given period and writes the results to a flat file.
- **Extract Tab-Delimited Initial Load/Ongoing Consumption (with UTC Offset)** (D2-IDEXTPRDU): This algorithm retrieves a service point's consumption for a given period and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.
- **Extract Initial/Ongoing Consumption and Apply TOU Map for DataConnect** (D2-IDEXTPTOU): This algorithm retrieves a service point's consumption for a given period, applies a TOU Map to the consumption, and writes the results to a flat file.
- **Extract Tab-Delimited Initial/Ongoing Consumption and Apply TOU Map (with UTC Offset)** (D2-IDEXTPTUU): This algorithm retrieves a service point's consumption for a given period, applies a TOU Map to the consumption, and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.


Copyright © 2026, Oracle and/or its affiliates.

**Historical Corrections**: the following algorithms are used to extract and format interval data for historical correction requests:

- **Extract Historical Correction Consumption for DataConnect** (D2-IDEXTIMD): This algorithm retrieves historical correction consumption for a service point and writes the results to a flat file.
- **Extract Tab-Delimited Historical Correction Consumption (with UTC Offset)** (D2-IDEXTIMDU): This algorithm retrieves historical correction consumption for a service point and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.
- **Extract Historical Corrections and Apply TOU Map for DataConnect** (D2-IDEXTITOU): This algorithm retrieves historical correction consumption for a service point, applies a TOU map to the consumption, and writes the results to a flat file.
- **Extract Tab-Delimited Historical Corrections and Apply TOU Map (with UTC Offset)** (D2-IDEXTITUU): This algorithm retrieves historical correction consumption for a service point, applies a TOU map to the consumption, and writes the results to a tab-delimited flat file. This algorithm also supports adjusting the consumption based on Daylight Saving Time.

### Batch Controls

The consumption extraction process uses a set of base package batch controls to extract and format the interval data for export.

- **Create Daily Consumption Extract Requests** (D2-CRERQ): This batch process scans for active Consumption Extract Types, and for each one that has Frequency of Automated Daily creates a pending request. This process should be scheduled to run daily (or at another regular interval).

The following sample batch controls are provided to extract and format interval data. Unique batch controls of each of these is required for each consumption extract type. You should create custom versions of the above batch controls for each consumption extract type in your implementation. Extract type-specific versions of these batch controls should be specified in the "Batch Control" section on the Consumption Extract Type as appropriate.

- **Initial Load/Ongoing Consumption Extract** (D2-IDEPD): This batch process extracts interval data for a specified period. This batch control uses the "Initial Load/Ongoing Extract Algorithm" defined on the Consumption Extract Type.
- **Historical Corrections Consumption Extract** (D2-IDEHC): This batch process extracts interval data for historical corrections. This batch control uses the "Historical Corrections Extract Algorithm" defined on the Consumption Extract Type.

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path or object storage location and file name for this file.) for a delimited flat file containing extracted data. Refer to External File Storage for more information about setting up object storage locations.

### Example Setup Steps

Setting up consumption extracts involves the following steps:

1. Create Consumption Request Types: you should create a consumption request type for each unique type of request required by your implementation.
2. Create Consumption Extract Types: you should create a consumption extract type for each unique combination of output details (target UOM, target interval size, TOU map/template) and measurement selection criteria.


Copyright © 2026, Oracle and/or its affiliates.

3. Add the "Create General Process Record if IMD is Historical Correction" historical correction algorithm as an Enter algorithm on the Finalized state of the initial measurement business objects.
4. Add the "Create General Process Record for Re-derived Values" historical correction algorithm as an Enter algorithm on the Finalized state on the measurement business object.
5. Create and submit initial load consumption extract requests for each consumption extract type you created earlier.
6. Set up batch processes for daily extract requests and historical corrections. Batch processing for consumption extracts should include the following:
    a. The **Create Daily Consumption Extract Requests** (D2-CRERQ) batch control should be configured to run on a regular (i.e. daily) schedule to create ongoing consumption extract requests.
    b. The **Request Monitor (Deferred)** (F1-SUBRQ) batch control should be used to monitor for pending requests and transition them to the "Submit Job" state.
    c. Historical corrections consumption extract batch controls (based on D2-IDEH, one for each consumption extract type), should be configured to run on a regular basis to check for and create historical correction extracts.

# Master Data Extract

Extraction of service point, device, and install event master data is performed through use of data synchronization, audit algorithms, business objects and related batch processes.

Batch processes are used to create initial load extracts for service point, device, and install event master data. Following the initial load, data synchronization requests are created when master data is changed.

### Maintenance Object Audit Algorithms

Detecting changes in service point, device, and install event master data can be detected via audit algorithms on the Service Point (D1-SP), Device Configuration (D1-DVCCONFIG), Device (D1-DEVICE),and Install Event (D1-INSTLEVT) maintenance objects. The Generic Change Data Capture (F1-GCHG-CDCP) algorithm type provided in the base package can be used to create audit algorithms for the Service Point and Install Event maintenance objects.

In addition, the "Device Change Data Capture (Install Event-Based)" (D1-IEDV-CDCP) algorithm type provided in the base package can be used to create an audit algorithm on the Device maintenance object to determine if an install event sync request record is to be created. A change in a related device's details will instantiate an Install Event Extract Sync Request (of the type defined for the "Install Event Sync Request BO" algorithm parameter) if one does not already exist in the initial state for the Install Event.

### Business Objects and Algorithms

The maintenance object audit algorithms create data synchronization requests based on the "Sync Request BO" maintenance object options. Extraction of service points and install events are supported by the following data synchronization business objects.

- Current DataConnect
    - **SP/Meter Sync for DataConnect**(D1-ExternalRepositorySPMtrSync): used to extract service point and meter information. This business object should be defined as a value


Copyright © 2026, Oracle and/or its affiliates.

for the "Sync Request BO" options on the Service Point, Device Configuration, and Device maintenance objects.

- Legacy DataConnect
  - **SP Sync for Legacy DataConnect** (D1-ExternalRepositorySPSync): used to extract service point information. This business object should be defined as a value for the "Sync Request BO" options on the Service Point maintenance object.
  - **Install Event Sync for Legacy DataConnect** (D1-ExternalRepositoryIESync): used to extract install event information. This business object should be defined as a value for the "Sync Request BO" options on the Install Event maintenance object.

These business objects use the following Pre-Processing algorithms to take initial data snapshots, and define the batch control used to extract data and export to a flat file:

- Legacy DataConnect
  - **Capture SP Initial Snapshot for Legacy DataConnect** (D1-SPEINISNP)
  - **Capture SP/Meter Initial Snapshot for Legacy DataConnect** (D1-SPMINISNP)
  - **Capture Install Event Initial Snapshot for Legacy DataConnect** (D1-IEEINISNP)

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for related synchronization requests, and transitions them to the "Determine if Sync Needed" state. Enter algorithms on the "Determine If Sync Needed" states extract a final snapshot of the data to extract and export.

- Current DataConnect
  - **Capture SP/Meter Final Snapshot for DataConnect** (D1-SPMFINSNP)
- Legacy DataConnect
  - **Construct SP Final Snapshot for Legacy DataConnect** (D1-SPEFINSNP)
  - **Capture Install Event Final Snapshot for Legacy DataConnect** (D1-IEEINISNP)

The "Prepare Delimited Extract Data" Enter algorithm (D1-PRPEXTDTA) on the "Send Request" state prepares the data for extraction, and creates a general process record for the synchronization request (based on the batch control defined by the pre-processing algorithm). Note that this algorithm is defined on the "Send Request" state of the Generic Sync for DataConnect (D1-ParentExternalRepositorySyn) parent business object.

### Batch Controls

The master data extraction process uses a set of batch processes to create data synchronization requests and extract files. The following batch processes are used to create initial synchronization requests:

- Current DataConnect
  - **SP/Meter Initial Load for DataConnect** (D1-SPMIL): This batch control creates initial synchronization requests for service points, device configurations, and devices.
- Legacy DataConnect
  - **SP Initial Load for Legacy DataConnect** (D1-SPEIL): This batch control creates initial synchronization requests for service points.
  - **Install Event Initial Load for Legacy DataConnect** (D1-IEEIL): This batch control creates initial synchronization requests for install events.


Copyright © 2026, Oracle and/or its affiliates.

These batch processes should be run to create initial load data synchronization requests based on the "Sync for DataConnect" business objects. The following batch processes are used to create extract files from synchronization requests.

- Current DataConnect
  - **SP/Meter Extract for DataConnect** (D1-SPMSR): This batch control creates extract file(s) that contain service point, device configuration, and device information.
- Legacy DataConnect
  - **SP Extract for Legacy DataConnect** (D1-SPESR): This batch control creates extract file(s) that contain service point information.
  - **Install Event Extract for Legacy DataConnect** (D1-IEESR): This batch control creates extract file(s) that contain install event information.

These batch controls are defined as values for the "Batch Control for Extract" algorithm parameters on the Pre-Processing algorithms on the "Sync for DataConnect" business objects (see above).

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name) for a delimited flat file containing extracted data.

### Example Setup Steps

Setting up master data extracts involves the following steps:

1. Add audit algorithms on the Service Point and Install Event maintenance objects.
2. Add the DataConnect synchronization request business objects as "Sync Request BO" options on the Service Point and Install Event maintenance objects.
3. Add the "Install Event's Device Change Data Capture" algorithm as an audit algorithm on the Device maintenance object.
4. Execute initial load batch processes for service points and install events.
5. Once the initial load synchronization has been executed, changes to service points, install events, or related devices will trigger the creation of new synchronization requests and resulting extraction files.

## Extract Flat File Formats

The data included in each file is based on a specific Data Area. For each Data Area, this section provides a table that lists the name and corresponding schema element and metadata field for each data element extracted by the Data Area. Reference the Data Areas below to see the details for the fields included in each output.

| Process | Description | Data Area |
|---|---|---|
| **Install Event Snapshot for Legacy DataConnect** | The Install Event Snapshot for DataConnect data area is used for install event master data extract. | D1-ExternalRepositoryIESnapsht |
| **SP Meter Snapshot for DataConnect** | The Service Point and Meter Snapshot for DataConnect v1+ data area is used for service point and meter master data extract. | D1-ExternalRepositorySPMtrSnap |


Copyright © 2026, Oracle and/or its affiliates.

| Process | Description | Data Area |
|---|---|---|
| **Service Point Snapshot for Legacy DataConnect** | The Service Point Snapshot for DataConnect data area is used for service point master data extract. | D1-ExternalRepositorySPSnapsht |
| **Extract Snapshot for DataConnect (Condition)** | The Consumption Extract Snapshot data area is used for interval consumption extract, with measurement conditions. | D2-IntDataExtSnapshotMsrmtCond |
| **Extract Snapshot for Legacy DataConnect** | The Consumption Extract Snapshot data area is used for interval consumption extract. | D2-IntervalDataExtRepoSnapshot |
| **Extract Snapshot for Legacy DataConnect (TOU Mapping)** | The Consumption Extract Snapshot (TOU Mapping) data area is used for interval consumption extract mapped to TOU periods. | D2-IntervalDataExtRepoTOUSnap |
| **Extract Snapshot for DataConnect** | The Consumption Extract Snapshot data area is used for interval consumption extract, with UTC offset. | D2-IntervalDataExtRepoSnpstUTC |
| **Extract Snapshot for DataConnect (TOU Mapping)** | The Consumption Extract Snapshot (TOU Mapping) data area is used for interval consumption extract mapped to TOU periods, with UTC offset. | D2-IntervalDataExtRepoTOUSnUTC |

Specifics for how the flat files are created are defined on batch controls and algorithms used by the extract process.

- **File Name and Path**: Parameters on batch controls define the file name, path, and other details about the output file.
- **File Size and Contents**: Batch controls for initial load/ongoing consumption requests include a parameter to specify the number of service points to be included in each file.
- **Character Encoding and File Delimiter**: Parameters on the Prepare Delimited Extract Data (D1-PRPEXTDTA) algorithm are used to specify the character encoding and delimiter used in the flat files for master data extracts.

### Example Extract File

The following example illustrates comma-separated interval data extracts based on the Consumption Extract Snapshot data area:

19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,3,600,1.366,2015-01-01,07.00.00,No

19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,3,600,1.366,2015-01-01,08.00.0,No

19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,3,600,1.366.2015-01-01,09.00.0,No

19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,3,600,1.366,2015-01-01,10.00.00,No


Copyright © 2026, Oracle and/or its affiliates.

19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,3,600,1.366,2015-01-01,11.00.00,No

19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,3,600,1.366,2015-01-01,12.00.00,No

19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,3,600,1.366,2015-01-01,13.00.00,No

# Oracle Utilities Data Intelligence


- Defining Standard Measurement Identifiers


## Defining Standard Measurement Identifiers

Oracle Utilities Data Intelligence (OUDI) can perform data aggregation and enrichment on Advanced Metering Infrastructure (AMI) data extracted from Oracle Utilities Meter Solution Cloud Service and Oracle Utilities Customer Cloud Service. Data is extracted using Measurement Data Extract.

Enabling the extract process requires defining the data to extract which involves defining the "standard measurement identifiers" used by Data Intelligence in Meter Solution Cloud Service and Customer Cloud Service, and mapping those identifiers to specific combinations of Units of Measure (UOM), Time of Use (TOU), and Service Quantity Identifiers (SQI) for specific service types.

### Extendable Lookups

The Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup is used to define standard measurement identifiers. Values for this extendable lookup include the following:

- **Standard Measurement Identifier**: A code for the identifier
- **Standard Service Type**: The basic or standard service type for the identifier, from the Standard Service Type (STD_SVC_TYPE_FLG) lookup.
- **Status**: The current status of the identifier
- **Description**: The label used for the identifier in the Meter Solution Cloud Service and Customer Cloud Service user interface.

As an example, the standard measurement identifier for consumed electric kilowatt hours is defined as follows:

- **Standard Measurement Identifier**: E-KWH-CONS
- **Standard Service Type**: Electric
- **Status**: Active
- **Description**: Electric kWh - Consumed

### Master Configurations

A set of "Measurement Identification Configuration" master configurations are used to map relationships between "standard measurement identifiers" defined in the Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup and specific UOM/TOU/SQIs combinations per service type. There are three of these master



Copyright © 2026, Oracle and/or its affiliates.

configurations, one for each of the Standard Service Types used in the extendable lookup. These are:

- Measurement Identification - Electric Configuration (D1-MsrmtIdentifyElectricConfig)
- Measurement Identification - Gas Configuration (D1-MsrmtIdentifyGasConfig)
- Measurement Identification - Water Configuration (D1-MsrmtIdentifyWaterConfig)

Mappings in these master configurations are defined by the following:

- **Service Type**: The service type associated with a channel (see Service Types for more information)
- **Channel Identification**: Details regarding how to identify data for a channel, including
  - **Standard Measurement Identifier**: One or more standard measurement identifiers (from the Standard Measurement Identifier (D1-StdMsrmtIdentifierLookup) extendable lookup) for the channel
  - **Color Option**: The color used to display measurement data for this channel (from the Color Option (F1-Color) extendable lookup)
  - **Measurement Value Identification**: One more UOM/TOU/SQI combinations that defines what the channel measures
    - **Unit of Measure**: The unit of measure for the combination, such as kWh, kW, and so on
    - **Time of Use**: The time of use period for the combination, such as On-Peak, Off-Peak, and so on
    - **Service Quantity Identifier**: The service quantity identifier for the combination, such as "Generated", "Consumed", and so on

For example, the following configuration could be used to map three standard measurement identifiers (Electric kWh - Consumed, Electric kWh - Generated, and Electric kW) to the "Electric Services" service type and define the UOM/TOU/SQI combinations for each.

- **Service Type**: Electric Service
- **Channel Identification**:
  - **Standard Measurement Identifier**: Electric kWh - Consumed
  - **Color Option**: Green 010
  - **Measurement Value Identification**:
    - **Unit of Measure**: Kilowatt Hour
    - **Service Quantity Identifier**: Consumed
  - **Measurement Value Identification**:
    - **Unit of Measure**: Kilowatt Hour
- **Standard Measurement Identifier**: Electric kWh - Generated
  - **Color Option**: Blue 010
  - **Measurement Value Identification**:
    - **Unit of Measure**: Kilowatt Hour
    - **Service Quantity Identifier**: Generated
- **Standard Measurement Identifier**: Electric Kilowatt
  - **Color Option**: Orange 010


Copyright © 2026, Oracle and/or its affiliates.

- **Measurement Value Identification**:
  - **Unit of Measure**: Kilowatt

# Message Formats

Oracle Utilities Meter Data Management supports three core transactional data imports: Initial Measurement Data, Device Event Data, and Usage Transactions. To see the appropriate format for each of these imports please see the following:

- **Initial Measurement Data**: Please see the IMD Seeder (D1-IMDSeeder) business object for the appropriate format
- **Device Events**: Please see the Device Event Seeder (D1-DeviceEventSeeder) business object for the appropriate format
- **Usage Transactions**: Please see the Usage Transaction Seeder (D2-UsgTranSeeder) business object for the appropriate format*

# Data Import - Message Driven Bean Configuration


- Overview
- JMS Configuration
- Message Driven Bean Configuration
- Notification Queue Configuration


## Overview

This section describes the steps for configuring the Message Driven Bean (MDB) feature of Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway to listen to inbound JMS messages. This feature is used when importing IMDs and device events from head-end systems or when receiving Usage Transactions from CIS.

## JMS Configuration

JMS configuration involves setting up JMS queues which will received inbound usage readings and device events. The JMS queues need to be created first on the application server where the OSB component is deployed. This server is referred to as remote server in the sections below. In the following section the JMS queue on the remote server is assumed to be created with the name **DestinationQueueWatch-CM**.

Note: The JMS changes described in the following sections are not persistent during patches or upgrades. They will need to be re-created after applying any patches or upgrades to Oracle Utilities Smart Grid Gateway. It is recommended to keep a backup of the `$SPLEBASE/splapp/config.xml` file.

#### Create a new JMS Module

Log in to the Oracle Utilities Smart Grid Gateway Weblogic console and create a JMS Module with an appropriate name. Specify the following values for this JMS module:

- **Name**: the name of JMS module. For example, JMSModule-CM



Copyright © 2026, Oracle and/or its affiliates.

- **Target**: the name of the target server where the Oracle Utilities Smart Grid Gateway application is running. This should be specified as myserver.

#### Create a Foreign JMS Server

Create a Foreign JMS server under the JMS module created in the above step. Specify the following values for this foreign JMS server:

- **Name**: Name of the foreign server. For example, JMSFAServer-CM
- **Target**: This should be specified as myserver
- **JNDI Initial Context Factory**: This should be specified as weblogic.jndi.WLInitialContextFactory
- **JNDI Connection URL**: The URL of the server where OSB is deployed. For example: t3://osbserver:7001
- **JNDI Properties Credential**: Password for the OSB server user
- **JNDI Properties**: The java.naming.security.principal additional property should be specified and set to the OSB server user. For example, java.naming.security.principal=weblogic

#### Create a Foreign Destination

Create a Foreign destination for each remote queue. Specify the following values for this foreign destination:

- **Name**: Name of foreign destination. For instance, DestinationQueue-CM
- **Local JNDI Name**: Local JNDI name for the foreign JMS queue. For example, ForeignDestinationQueue-CM
- **Remote JNDI Name**: JNDI name of the queue on the remote server. For example, DestinationQueueWatch-CM

#### Create a Remote Connection Factory

Create a remote connection factory for the foreign JMS server. Specify the following values for this remote connection factory:

- **Name**: Name of remote connection factory. For example, DestinationQueueConnectionFactory-CM
- **Local JNDI Name**: Local JNDI name for the Remote Connection Factory. For example, ForegnDestinationQueueConnectionFactory-CM
- **Remote JNDI Name**: JNDI name of the JMS Connection Factory on the remote server. For example, weblogic.jms.XAConnectionFactory

## Message Driven Bean Configuration

Configuration of message driven beans (MDB) involved modifying the **ejb-jar.xml** and **weblogic-ejb-jar.xml** configuration files delivered with Oracle Utilities Smart Grid Gateway. It is recommended that instead of modifying these files directly you create "Customer Modification" (CM) versions of these files to make changes to these configuration files. This ensures that your modifications are not overwritten by future application patches.

The following section describes the changes required in the CM files for configuring the MDBs to read from the foreign JMS queues set up in the steps above. This requires creating the following files under $SPLEBASE/templates:


Copyright © 2026, Oracle and/or its affiliates.

- cm_ejb-jar.xml.wls.jms_1.include
- cm_ejb-jar.xml.wls.jms_2.include
- cm_weblogic-ejb-jar.xml.jms.include.

**Note**: After making these changes the initialSetup script needs to be run and Oracle Utilities Smart Grid Gateway application needs to be redeployed. However the initialSetup script will overwrite the JMS configuration changes made in the steps above. So it is recommended to keep a backup of the $SPLEBASE/splapp/config.xml file before running this script.

**Changes to cm_ejb-jar.xml.wls.jms_1.include**

The following is an an example of the `cm_ejb-jar.xml.wls.jms_1.include` file:

```
<message-driven>
   <description>MDB for DestinationQueue-CM</description>
   <display-name>DestinationQueueWatcher-CM</display-name>

      <ejb-name>DestinationQueueWatch-CM</ejb-name>

   <ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
   <messaging-type>javax.jms.MessageListener</messaging-type>
   <transaction-type>Bean</transaction-type>
   <message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
```

The values specified in the above file include the following:

- **ejb-name**: This is the name of the MDB.

**Changes to cm_ejb-jar.xml.wls.jms_2.include**

The following is an example of the `cm_ejb-jar.xml.wls.jms_2.include` file:

```
   <assembly-descriptor>
   <security-role>
   <role-name>cisusers</role-name>
   </security-role>
   <container-transaction>
   <method>

      <ejb-name>DestinationQueueWatch-CM</ejb-name>

   <method-name>onMessage</method-name>
   </method>
   <trans-attribute>NotSupported</trans-attribute>
   </container-transaction>
   </assembly-descriptor>
```

The values specified in the above file include the following:

- **ejb-name**: This is the name of the MDB


Copyright © 2026, Oracle and/or its affiliates.

**Changes to cm_ejb-jar.xml.wls.jms_2.include**

The following is an example of the `cm_weblogic-ejb-jar.xml.jms.include` file:

```
<weblogic-enterprise-bean>

 <ejb-name>DestinationQueueWatch-CM</ejb-name>

<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>

 <destination-jndi-name>ForeignDestinationQueue-CM</destination-jndi-name>

 <connection-factory-jndi-name>ForeignConnectionFactory-CM</connection-
factory-jndi-name>

</message-driven-descriptor>
</weblogic-enterprise-bean>
```

The values specified in the above file include the following:

- **ejb-name**: This should be the name of the MDB as specified in ejb-jar.xml
- **destination-jndi-name**: This should be the JNDI name of the foreign destination as provided in JMS module ' Foreign server ' Foreign destination ' Local JNDI name.
- **connection-factory-jndi-name**: This should be the JNDI name of the connection factory as provided in JMS module ' Foreign server ' Remote Connection Factory ' Local JNDI name.

# Notification Queue Configuration

Payload statistics and payload summary records must be submitted sequentially in order for them to be processed correctly. To prevent them from being processed at the same time, you should set the number of notification queue polling threads to 1. Follow these steps to configure the number of notification queue threads:

1. Log in to the WebLogic Server Administration Console.
2. Under Helpful Tools, click **Configure Applications.**
3. Click on **Webservices**.
4. Click on the NotificationQueue link. for the EJB that you want to configure.
5. Go to the Configuration tab.
6. In the Change Center, click **Lock & Edit.**
7. Specify the new value of polling threads in **Max Beans in Free Pool.**
8. Click **Save.**
9. Click **Release Configuration.**
10. Restart the OUAF WebLogic instance.


Copyright © 2026, Oracle and/or its affiliates.

# Creating Application Links in External Applications

When integrating Oracle Utilities applications with other external applications, you can create links from external applications to specific portals and pages within the Oracle Utilities applications (also known as "deep linking"). These links typically leverage Business Process Assistant (BPA) scripts that direct the user to the appropriate target portal or page for a specific entity such as an account or service point. Refer to Creating Application Links in External Applications in the *Application Framework Administrative User Guide* for more information about creating links.

**Note**: When creating application links, the use of the following parameters will help speed up launching the target page

- **initNav=false**: skips launching the default home page for the user
- **minSidebar=true**: minimizes the Sidebar for the user

### Go To Portal Script

The Framework provides a generic script (F1-GoToPrtl) that can be used to navigate to the maintenance portal for a specified maintenance object and entity. Refer to Navigating to a Given Record's Maintenance Portal in the *Application Framework Administrative User Guide* for more information about using this script.

### Go To MDM Script

Oracle Utilities Meter Data Management includes the Go To MDM script (D1-GoToMDM) that can be used to navigate directly to a specific 360 Degree View tab. This script uses the following syntax:

```
script=D1-GoToMDM&FLAG=<FLAG_VALUE>&PKVALUE1=<ID>
```

where:

- **<FLAG_VALUE>**: A value from the DRILL_ENTITY_FLG lookup that specifies which 360 Degree View page to open.
- **<ID>**: The primary key value for the entity (such as a Device, Measuring Component, or Service Point) you wish to view.

**Examples**:

| Script | Flag Value | ID | Example |
|---|---|---|---|
| D1-GotoMDM | FLAG=D1DV<br>(360 View for Device) | Device ID | script=D1-GotoMDM&FLAG=D1DV&PKVALUE1={your Device ID} |
| D1-GotoMDM | FLAG=D1MC<br>(360 View for Measuring Component) | Measuring Component ID | script=D1-GotoMDM&FLAG=D1MC&PKVALUE1={your Measuring Component ID} |


Copyright © 2026, Oracle and/or its affiliates.

| Script | Flag Value | ID | Example |
|---|---|---|---|
| D1-GotoMDM | FLAG=D1SP<br>(360 View for Service Point) | Service Point ID | script=D1-GotoMDM&FLAG=D1SP&PKVALUE1={your Service Point ID} |

**Full URL Example**:

```
https://<ENV_BASE_URL>/ouaf/cis.jsp/?minSidebar=true&initNav=false&script=D1-GotoMDM&FLAG=D1DV&PKVALUE1={your Device ID}
```


Copyright © 2026, Oracle and/or its affiliates.

# Part V
# Oracle Utilities Cloud Service Foundation

Welcome to the Oracle Utilities Cloud Service Foundation Administrative User Guide. This document describes how to administer features associated with Oracle Utilities Cloud Service Foundation.

> **Note**
>
> This document does not apply to all implementations. Contact your system administrator to confirm if your implementation includes Oracle Utilities Cloud Service Foundation.


- Process Automation Tool
- Data Conversion Support for Cloud Implementations
- Batch Scheduler for Cloud Implementations
- Batch Queues Overview
- Batch Alerts
- ILM Dashboard for Cloud Implementations
- Oracle Cloud Infrastructure Identity and Access Management Integration
- Oracle Cloud Object Storage
- Maintaining Customer Code


# Process Automation Tool

The Process Automation Tool allows customers and implementers to orchestrate and automate a set of infrastructure related multi-step processes. The tool provides the ability to monitor the progress of such process and each of the steps. The tool also supports user actions at the process and step level.

The process automation tool is made out of the following system entities:

- Infrastructure Process - a Business Object of the Service Task MO.
- Infrastructure Process Step - a Business Object of the Service Task MO.
- Infrastructure Process Type - a Business Object of the Service Task Type MO.
- Infrastructure Process Step Type - a Business Object of the Service Task Type MO.

An Infrastructure Process references a type that includes the steps that are a part of that process. The steps are executed one by one according to their sequence as long as the previous step ended successfully. When a step in a process fails, the process will stop and wait for a user to take further actions.



Copyright © 2026, Oracle and/or its affiliates.

An Infrastructure Process can span up to two product environments. This means that it can invoke and monitor steps in two systems. This, for example, allows it to orchestrate a configuration migration (using the Content Migration Assistant, described in your cloud service's *Administrative User Guide*) from one product environment to another. Such a migration can include steps to create the export data on the **source** environment, move the export file to the **target** environment and execute all the import tasks on the target environment, all as a single Infrastructure Process. In this case the Infrastructure Process will be created on the **target** environment and will pull the configuration from the **source** environment.

> **Note**
>
> This is done in order to ensure that only application users with sufficient security privileges in any system environment can create Infrastructure Processes that can update that environment.
>
> The rule is that when an Infrastructure Process is created to run processes on two systems, the process will always be created on the **environment with the higher security requirements**. For example, if you need to run processes on the Development and Test environments, the infrastructure process will be created and executed from the Test environment (assuming that the Test environment has higher security requirements than the Development environment).

The steps that make an Infrastructure Process reference a step type that governs their behavior. A comprehensive set of process and step types is provided with the product. This set can be extended by implementers as needed.

- Creating an Infrastructure Process
- Executing an Infrastructure Process
- Monitoring Infrastructure Process Progress
- Process Automation Tool Setup
- Configuring New Infrastructure Processes
- CMA Migration Infrastructure Process Types

# Creating an Infrastructure Process

To create a new process, navigate using **Admin > Implementation Tools > Infrastructure Process**. Use the Infrastructure Process Query to search for an existing process. One an Infrastructure Process is selected, you are brought to the maintenance portal to view and maintain the selected record.

The following points provide information about creating a new Infrastructure Process.

Infrastructure Process references an **Infrastructure Process Type**. When creating a new process you need to select the type that the process will be based on.

The **Name** of the process is used for tracking purposes. The name doesn't have to be unique in the system.

The **Process Steps** list is read-only and represents the steps that were configured on the Infrastructure Process Type.

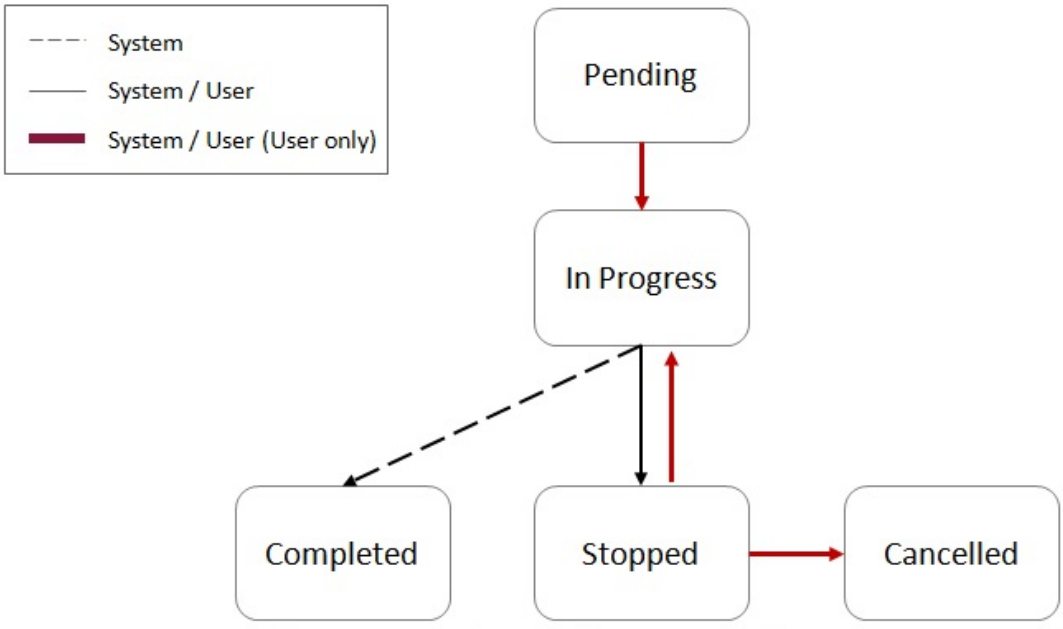

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Changing information on the Infrastructure Process Type, including the steps, will not affect existing Infrastructure Process records.

The **Process Details** section represents additional information that is required according to the corresponding process type definition.

# Executing an Infrastructure Process

**Infrastructure Process Lifecycle**




New processes are created in a Pending status. While in this status the Infrastructure Process record can be modified, deleted or duplicated.

> **Note**
>
> Once an Infrastructure Process record is created, its type cannot be changed. To change the type, you can delete the process that was created with the wrong type (if it is still in Pending status) and create a new process with the correct type.

Use the **Start** action, on the maintenance portal to begin the process execution. The process status will change to **In Progress**.

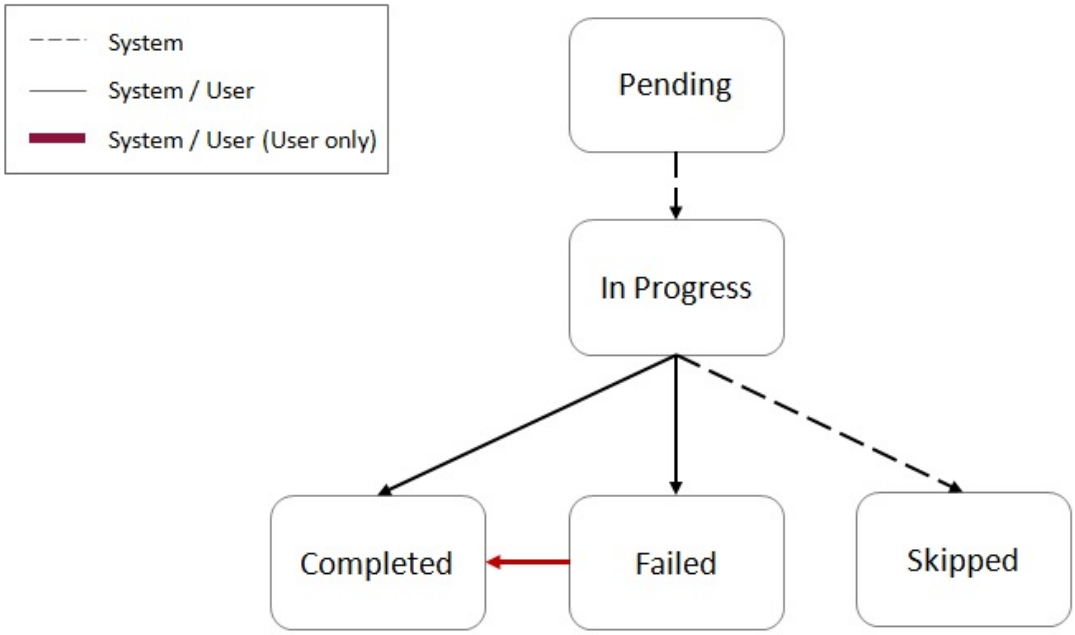

Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> While the Infrastructure Process is in **Pending** status, the steps will not have a status or any available actions. When the process transitions to **In Progress** status the process step records are created. From that point on they have a status and actions according to the step that and status.

While the process is **In Progress** the steps of the process will execute one by one according to their sequence. The process can be stopped using the **Stop** action. In this case the current step that is in progress will continue but the next step will not be executed. If any of the steps fail, the process will also transition into **Stopped** status and will not execute the next step.

When the process is in **Stopped** status it can be resumed by selecting the **Resume** action, or it can be cancelled by selecting the **Cancel** action.

**Infrastructure Process Step Lifecycle**




A process steps is created in **Pending** status and is transitioned to **In Progress** by the process logic when its turn to execute arrives.

When a step is **In Progress**, the following transitions are possible:

- Transition the step into Completed status upon successful completion of the step's work or by selecting the **Complete** action.
- Transition the step into **Failed** status when an error was detected during the step's work or by selecting the **Fail** action.
- Steps may include logic that will transition the step into **Skipped** status if that logic determines that the step should not be executed.

When the step is In Progress, selecting the **Job Status** action will display information about the job running status (if it is a batch job type of step).


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Manually changing the step status to **Completed** or **Failed** while the step is In Progress should be carefully considered. This should only be done when step stays **In Progress** status for a long time and upon investigation it is clear that some manual intervention is required.

When a step is in **Failed** status it is still possible to manually transition the step into Completed status.

> **Note**
>
> Transitioning the step from **Failed** to **Completed** status should only be done if it is clear that the step was able to successfully accomplish its task. This will likely be established after some investigation.

# Monitoring Infrastructure Process Progress


- Infrastructure Process Control
- Monitoring a Running Infrastructure Process
- Dealing with Exceptions


## Infrastructure Process Control

The Infrastructure Process Control portal provides an overview of Infrastructure Processes that are active (i.e. in **Pending** or **In Progress** status) or processes that did not complete (i.e. in **Stopped** status) within the last X days.

Navigate to Infrastructure Process Control portal, using **Admin > Implementation Tools > Infrastructure Process Control**.

Available actions on this portal are:

- **New Process** - creates a new Infrastructure Process
- **Environment Setup** - directs the user to the master configuration related to the process automation tool

Select any of the processes from the portal lists for more information.

## Monitoring a Running Infrastructure Process

When a process is running (in **In Progress** status), you can monitor its progress by refreshing the page. To configure automatic page refresh, refer to the User Portal Preferences for this portal.

The Infrastructure Process maintenance portal provides information about the status of the overall process as well as each of the steps.



Copyright © 2026, Oracle and/or its affiliates.

You can also view the details of each individual step including their log for more information or to investigate exceptions.

# Dealing with Exceptions

Infrastructure Process steps can be divided into two main categories:

- Steps that start a batch process (job)
- Steps that are not batch process related

In addition, steps can be local or remote. Local steps execute on the same product environment that the process is executing while remote steps are executed on a different product environment.


- Batch Job Step Exceptions
- Non-Batch Step Exceptions


## Batch Job Step Exceptions

Batch job steps start a batch job either locally or on a different product environment.

### Batch Job Step in Failed Status as a Result of a Batch Job Error

When a batch process (local or remote) ends with an error, the process step is automatically transitioned to **Failed** status.

In this case, you should:

1. Ascertain the cause of the job failure (in the local or remote environment).
2. If the issue is temporary or can be fixed, fix it and then rerun the batch job manually using the Batch Job Submission page (in the local or remote environment). If the resubmitted job ended successfully, you can transition the process step to the **Completed** status and use the **Resume** action at the process level to continue the process execution.

If the issue cannot be resolved, further investigation is required. In this case the process should remain in **Stopped** status.

### Batch Job Step in In Progress Status While Batch Job Ended With an Error

When an Infrastructure Process Step that is associated with a batch job stays in **In Progress** status while the batch job ended with an error, further investigation is required. One possible cause for that situation could be an error that causes the job to fail and at the same time also prevents the job from communicating its status back to the originating process step.

In this case you should manually transition the step into **Failed** status and continue your investigation.



Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If you decide to stop the Infrastructure Process altogether (as a result of the exception), it is a good idea to mark the batch job that failed "Do Not Attempt Restart" in the Batch Run Control for that batch run. This will ensure that the next time the Infrastructure Processes runs, when it gets to that batch process step it will not attempt to resume the execution that ended with an exception, possibly causing a repeat of the exception.

**Batch Job Step in In Progress Status While Batch Job Ended Successfully**

When an Infrastructure Process Step that is associated with a batch job stays in **In Progress** status while the batch job ended successfully, further investigation is required. In this case, once you ascertained that the job did actually end successfully, you can manually transition the process step into **Completed** status and use the **Resume** action at the process level to continue its execution.

## Non-Batch Step Exceptions

Infrastructure Process Steps that are not associated with batch jobs can experience errors as well and therefore can fail. As a reminder, the logic for these steps can experience local or remote errors. When such a step fails, information about the failure can be found in the process step log.

# Process Automation Tool Setup

The process automation tool requires the setup of the following components:

- Master Configuration
- External Systems
- Message Senders

In addition, a security setup is required.

The setup itself is done automatically as part of the environment provisioning process. The information in this section is provided for reference and in case changes in existing setup are required.

- Security Setup
- Master Configuration Setup
- External Systems and Message Senders

## Security Setup

Before the process automation tool can be used, the security credentials of the Message Senders, used by the process automation tool, must be updated.

In order to execute the security setup, navigate using **Admin > Implementation Tools > Process Automation Security Setup**.

The setup script will ask you to provide a User ID, a Password and to indicate whether or not to override the existing security setup.


Copyright © 2026, Oracle and/or its affiliates.

**Please contact your system or security administrator for more information about what User ID should be used for this setup.**

You should select to override existing security setup when you need to update the User ID or the Password on the existing Messages Senders, used by the process automation tool. A common use case will be mandatory periodic password rotation for the user account that was selected for the setup.

> **Note**
>
> At the end of the process, you will be asked to flush all the system cache. This is required in order for the information update to take effect.

# Master Configuration Setup

The process automation tool relies on master configuration settings in order to identify all the product environments that are available to it. The Master Configuration is automatically setup as part of the system provisioning process.

This section describes the setup options in case manual changes are needed.

To access the configuration for the process automation tool, navigate using **Admin** > **General** > **Master Configuration** and look for Process Automation Configuration.

The Main section includes the definition of the **Current Environment** code and the **default file extension** for CMA export files.

The Current Environment code is a **unique** code that is given to each product environment.

The Environment List section describes all the existing product environments that the process automation tool need to interact with, for example, in order to run a process that moves integration configuration from one environment to another.

The environment in the list includes the following details:

- Environment: the environment code.
- **Product**: defined in the Process Automation Product extendable lookup.
- **Domain**: defined in the Process Automation Domain extendable lookup.
- **Migration Source**: select "Yes" if you can migrate configuration data from this environment.

  > **Note**
  >
  > You should not allow automated migration of configuration from your development environment directly to your production environment but you would allow migrating from the development to test environments.

- **External System**: reference to the External System definition that will support communication with the row"s environment.

# External Systems and Message Senders


Copyright © 2026, Oracle and/or its affiliates.

The process automation tool communicates with other product environments through Outbound Messages.

The automated setup for the process automation tool creates a set of Message Senders and External Systems to support communication with other product environments.

Each product environment which is defined on the Process Automation Configuration has a corresponding unique External System. Each External System contains the details of all the message types that it supports, the message sender for each message type and additional data if required.

# Configuring New Infrastructure Processes

The process automation tool configuration is essentially a collection of process types and step types that support the automation of various multi-step processes. A set of Infrastructure Process Types and Infrastructure Process Step Types is provided with the base product. Customers and implementers can create new process and step types.

This section describes the actions required to configure new Infrastructure Process Type and Infrastructure Process Step Type.

- Configuring a New Infrastructure Process Step Type
- Configuring a New Infrastructure Process Type

## Configuring a New Infrastructure Process Step Type

In order to configure a new step type, navigate using **Admin > Implementation Tools > Infrastructure Process Step Type**. Use the Infrastructure Process Step Type Query to view existing step types.

> ⚠ **Caution**
>
> Base product step type names start with "K1" and should not be change. Changing these step types can break the process and step types that are provided with the product.

The following points provide information about creating a new Infrastructure Process Step Type.

When creating a new step type, a business object has to be selected first. The Infrastructure Process Step Type BO contains the information necessary for the Infrastructure Process Step that references that step type. For example, a batch process step type BO contains information related to the batch code that should be submitted and the parameters that will be passed to it. You can use existing Infrastructure Process Step Type BOs or create new BOs according to your business requirements.

The **Related Transaction BO** references the Infrastructure Process Step BO. The step BO contains the business logic associated with the step. For example, a batch process step BO is responsible to submit a batch job with to the information from the step type. You can use existing Infrastructure Process Step BOs or create new BOs according to your business requirements.

The **Process Step Type Details** section represents additional information that is required for this specific step type BO.


Copyright © 2026, Oracle and/or its affiliates.

## Configuring a New Infrastructure Process Type

In order to configure a new process type, navigate using **Admin > Implementation Tools > Infrastructure Process Type**. Use the Infrastructure Process Step all-in-one portal to view existing process types.

> ⚠ **Caution**
>
> Base product process type names start with "K1" and should not be change. Changing these process types can break the base product process types that are provided with the product.

The following points provide information about creating a new Infrastructure Process Type.

When creating a new process type, a business object has to be selected first. The Infrastructure Process Type BO contains information necessary for the Infrastructure Process that references that process type. You can use existing Infrastructure Process Type BOs or create new BOs according to your business requirements.

The **Related Transaction BO** references the Infrastructure Process BO. The process BO contains the business logic associated with the process. This includes the orchestration of the process steps. The process BO also contains information that is shared by step business logic during the process. For example, a process BO that handles migration of data between product environments can have a buffer that will hold the information pulled from the source environment using one step so that another step can use it to push that to the target environment. You can use existing Infrastructure Process BOs or create new BOs according to your business requirements.

The **Process Type Steps** section includes a collection of step types. The sequence of steps has to be unique and this will be sequence of execution.

> ⚠ **Caution**
>
> When selecting step types for a process type you need to consider the fact that not all step types can be used on all process types. For example, some steps need to get or save information in the process itself so it can be used by other steps of that process. These types of steps can only work with a certain type of process. Please refer to the detailed description in each of the base product provided Infrastructure Process Step Types to see what process types they can be referenced on.

The **Infrastructure Process Type Details** section represents additional information that is required for the specific process type BO.

# CMA Migration Infrastructure Process Types

The following Infrastructure Process Types are provided to support the following configuration migration use cases:

- CMA Accelerator Load - import an exported file created by the Content Migration Assistant export process.


Copyright © 2026, Oracle and/or its affiliates.

- CMA Migration - a wholesale or piecemeal configuration migration (using the Content Migration Assistant), exporting configuration from a source environment and importing it to a target environment.
- CMA Accelerator Load (K1-CMA-LOAD)
- CMA Migration (K1-CMA-MIGRATE)

# CMA Accelerator Load (K1-CMA-LOAD)

The K1-CMA-LOAD Infrastructure Process Type supports importing of an exported CMA file.

K1-CMA-LOAD is based on K1-CMAMigrationProcessType Infrastructure Process Type BO and uses the K1-CMAMigration as the Infrastructure Process BO.

The CMA Accelerator Load process type includes load an exported CMA and to run all the batch processes that are a part of the CMA import.

> **Note**
>
> This process type does not include a step for a manual user review. It is assumed that when loading an accelerator, there shouldn't be anything to review and the new configuration content is expected to be new for the most part or OK to override existing configuration.

### Using the CMA Accelerator Load Infrastructure Process Type

In order to import a CMA export file (e.g. an accelerator), you need to:

- Upload the file to the system designated storage location for CMA uploads. The location is configured on the CMA Master Configuration entry and is automatically defaulted to an Oracle Cloud Object Storage location during the system installation process.
- Create a new infrastructure process with the K1-CMA-LOAD Infrastructure Process Type and provide the following additional details:
  - **Name**.
  - The **Process Steps** list is for reference only.
  - **Export filename**: specify the CMA export file name that was uploaded to the system. You should specify the full name including the file extension name.
  - **User Review Needed**: select "Yes" only if you are expecting the CMA file to include data that already exists in the system and need to be reviewed before changes are applied.
  - **Default Status For Add**, **Default Status For Update** are relevant in case you selected to have a user review. In this case define the default status of migration data items when imported into the target environment.
- Once the process is saved, you can start it by selecting the Start action.

# CMA Migration (K1-CMA-MIGRATE)


Copyright © 2026, Oracle and/or its affiliates.

The K1-CMA-MIGRATE Infrastructure Process Type supports a full or partial migration of configuration data, from a remote environment (the source) to the environment that the process is running on (the target).

> **Note**
>
> The configuration migration process is done as a PULL (running on the target environment and pulling data from the source environment) to make sure that users without security privileges to make changes in a product environment "A" won"t be able to run a process from a remote environment "B" that will force changes into "A".

K1-CMA-MIGRATE is based on K1-CMAMigrationProcessType Infrastructure Process Type BO and uses the K1-CMAMigration as the Infrastructure Process BO.

This configuration migration process type includes all the necessary steps to create an export CMA file from a migration request on the source environment, store the file in a shared file location (accessible by the source and target environment processes) and to run all the import batch processes on the target environment to process the file. This process type also includes a step for a manual user review of the imported data if needed.

> **Note**
>
> The shared file location is configured on the Migration Assistant Configuration master configuration entity and is defaulted to an Oracle Cloud Object Storage location during the system installation process.

#### Using the CMA Migration Infrastructure Process Type

In order to migrate configuration data from a remote environment (the source environment) to the current environment (the target environment) you need to:

- Create a new infrastructure process with the K1-CMA-MIGRATE Infrastructure Process Type and provide the following additional details:
  - **Name**.
  - **From Environment**: select the source environment from which the data will be migrated to the target environment (the current environment). The list of valid environments to choose from is taken from the Process Automation Master Configuration. The target environment is always the current environment - the one that this process is created on.
  - The **Process Steps** list is for reference only.
  - **Migration Request**: specify the CMA Migration Request ID that will be used for the migration process. The migration request should exists on the source environment and will be validated against that environment.
  - **Export filename**: specify the CMA export file name that will be used for the export and import process. You should specify the full name including the file extension name. If you leave this field empty, the system will default the file name according to the migration request ID and the default file extension name.
  - **User Review Needed**: select "Yes" only if you are expecting the CMA file to include data that already exists in the system and need to be reviewed before changes are applied.


Copyright © 2026, Oracle and/or its affiliates.

  - **Default Status For Add**, **Default Status For Update** are relevant in case you selected to have a user review. In this case define the default status of migration data items when imported into the target environment.
- Once the process is saved, you can start it by selecting the Start action.

> **Note**
>
> Before using this Infrastructure Process Type, make sure that the import and export directories for the Migration Assistant Configuration master configuration entry are pointing to the SAME shared location. This is required for a successful completion of this process type.

# Data Conversion Support for Cloud Implementations

This chapter provides information and requirements on data conversion support for Oracle Utilities Cloud implementations.


- Data Conversion Overview
- Configuring Conversion Data Upload
- Input Data Files
- Conversion Orchestration
- Security Setup for Conversion
- Conversion Reconciliation Reports


## Data Conversion Overview

The ability to convert customer data from legacy systems to OUAF based products is critical for every implementation. The following chapter describes the overall process, tools and considerations that are specific to OUAF based product implementations on Oracle Cloud.


- Conversion Steps
- SQL Loader
- Conversion Artifacts


### Conversion Steps

In general data conversion effort in the cloud is divided into the following steps:

- Prepare for conversion
- Create conversion data files
- Upload conversion data files to the cloud
- Process conversion data files
- Process converted data in cloud application



Copyright © 2026, Oracle and/or its affiliates.

- Approve converted data for production mode

> **Note**
>
> It is important to note that the conversion process is a highly iterative process. All the steps above are likely to be done (completely or partially) multiple times during the conversion process.
>
> For example, customer typically start with a small set of data to convert, they fix issues discovered with that data and then re-create new data files and proceed to convert them. This can happen many times, each time with either a different data sample or bigger and bigger data conversion sets until all the required data has gone through the conversion process.

- Prepare for Conversion
- Create Conversion Data Files
- Upload Data Conversion Files to the Cloud
- Process Conversion Data Files
- Process Converted Data in Cloud Application
- Approve Converted Data for Production Mode

## Prepare for Conversion

Preparing for conversion will typically include the following actions:

- Identify the OUAF product entities (Maintenance Objects) for which data will be converted. Refer the specific OUAF product documentation for information about converted entities for that product. Typically, each of the product converted entities will have a set of conversion staging tables that will receive the data from your legacy system before they get processed into the system.

  > **Note**
  >
  > Conversion staging schema design may vary between products, please refer to specific product's documentation for more details.

- Evaluate your project's requirements for the Information Lifecycle Management (ILM) and Data Archiving. If your implementation uses ILM, the ILM Date must be provided as part of the legacy data extract for each eligible table.

  > **Note**
  >
  > ILM support may vary between products, please refer to specific product's documentation for more details.

- Identify tables and maintenance objects whose data requires special treatment. That may include high volume tables, extremely large CLOB data, tables with multiple CLOB fields or any other special considerations related to the legacy data being converted.


Copyright © 2026, Oracle and/or its affiliates.

Address the special circumstances by customizing the default configurations:

- Enable conversion for tables that are not marked for conversion in the metadata use the Master Configuration
- If your table(s) contain multiple CLOBs, but you don't plan to supply the legacy data for all of them, create Conversion Task Type for this table(s) and populate **Exclude CLOB Field** list and then add a corresponding entry to Master Configuration's Special Instructions
- If you dealing with high volumes of data, you may want to adjust the load parameters and options for the SQL Loader. In this case, create a **custom Control File Header** (Managed Content), configure a dedicated Conversion Task Type referencing this custom Control File Header and then add a corresponding entry to Master Configuration's Special Instructions. Note that creating the custom Control File Header will result in a control file composed from the custom fragment and the generated fields list.
- If the generated field definitions list in the control file doesn't satisfy your requirements, you may completely override the control file. In this case, create a **custom Control File** (Managed Content, same as above), configure a dedicated Conversion Task Type referencing this custom Override Control File and then add a corresponding entry to Master Configuration's Special Instructions. Make sure your custom Control File contains all the run-time substitution variables that appear in the base Control File Header.

See Conversion Task Types and Conversion Master Configuration chapters for more details.

- Generate the conversion artifacts for the product entities you plan to convert data for. This will include, for example, a set of data specification files that describe the format of data required for each file.
- Explore conversion tasks generated for tables and/or maintenance objects, examine the input data specifications and file lists and determine the expected data formats and input file names.
- Identify your various sources of data and extraction methods.

## Create Conversion Data Files

In this step you create the set of data conversion files that were identified for the conversion process. You can choose to generate data files for each staging table or create a smaller set for each converted Maintenance Object.

The data conversion files typically do not include all the data to begin with but start small and grow with every iteration of the conversion process until they reach their maximum size at the end.

> **Note**
>
> When performing data conversion in the cloud, customers don’t have access to the database staging tables. Data scrubbing and transformation activities has to take place before the data files are generated (in each conversion iteration) so the result of these activities is reflected on the data files that are moved to the cloud to be loaded into the staging tables.


Copyright © 2026, Oracle and/or its affiliates.

## Upload Data Conversion Files to the Cloud

Since data conversion files are created locally they have to be uploaded to the cloud environment in order to be processed and loaded into the conversion staging tables.

In this step you should setup a dedicated input location on the object storage and upload the input data files there.

For more details refer to File Locations .

## Process Conversion Data Files

After uploading the files to the cloud, you should run a set of batch processes that will process the data files using Oracle SQL Loader and insert the data to the product conversion staging tables.

The staging table load process also produces output files that provide feedback on the load process.

You can download these output files for review.

> **Note**
>
> The process that imports the data in the files to the staging tables, using Oracle SQL Loader, is using the artifacts that were generated in the "Prepare for Conversion" step.

For more details refer to Understanding Load Data Process .

## Process Converted Data in Cloud Application

When the data intended for conversion (for a conversion iteration) is loaded into the appropriate conversion staging tables, the OUAF application in the cloud can process that data according to the product's data model, technical and business rules.

Refer to the OUAF product specific documentation for further guidelines and information on the specific processes and guidelines related to data conversion.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> This is one step of the overall iterative process but it is also includes many internal steps, for example:
>
> 1. Validation of the data in the conversion staging tables.
> 2. Generation of the system assigned keys for the converted entities.
> 3. Migration of the converted data from the staging tables into the actual system tables.
>
> When the processing of the converted data in this iteration is completed, information from the system is used to identity changes necessary to source data, the extraction process or other activities involved in producing the data to be converted.
>
> The process can then start over by loading a new set of data into data conversion files, load them to the staging tables in the cloud and process them once more to get more feedback.

## Approve Converted Data for Production Mode

When the quality and scope of data converted has reached its target, the process can stop. Once the overall data has been approved for production mode the conversion process comes to an end. The process of approving the data includes many activities, for example:

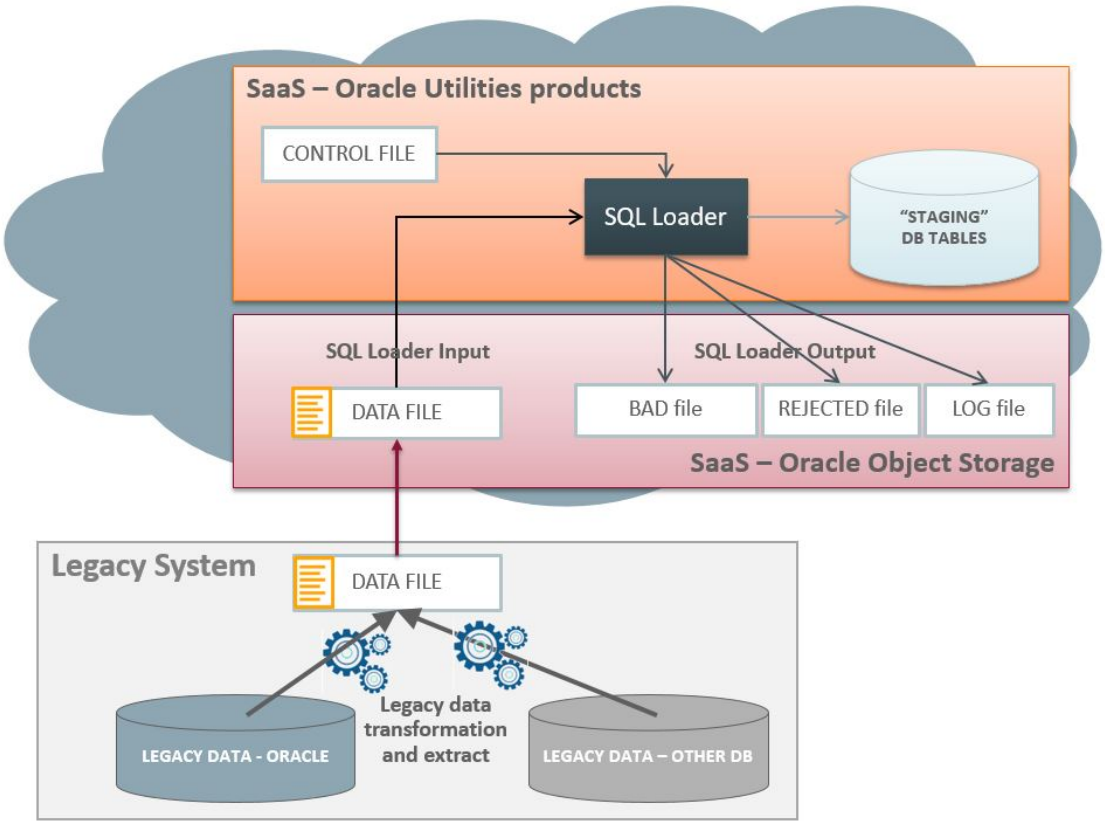
- Running various reports to compare converted data with legacy data to verify the quantity and quality of the data.
- Running reports and tests in the new application to verify the data and the system's behavior using that data.
- Manual review of random and key data in the system.

# SQL Loader

Loading the data in the data conversion files into the product conversion stating tables is done using Oracle SQL Loader.

Oracle SQL Loader is an Oracle Database Tool that enables mass upload of data from various sources into database tables in a fast and safe way.


Copyright © 2026, Oracle and/or its affiliates.




SQL Loader processing is determined by Control Files. The control files for the data conversion process are provided for each of the product conversion staging tables as part of the artifact generation which is a part of the preparations activities before starting the conversion process.

The Control Files typically define:

- The location of input files to load.
- The character set of the input data.
- The target table(s) for the load and how to insert the data (e.g. append to or override).
- How to map the data in the input file to tables and fields.
- The location of the output files (e.g. log, bad or rejected records).
- Options for parallel processing and load balancing.

The control files that are generated for the product include all the parameters necessary to load the product conversion staging tables. There are some customization options for these files that are discussed later.

After loading the control files, SQL Loader reads the records in the input data file(s), converts the input into database tables and columns that are inserted into the database.

SQL Loader has 2 modes of operation:

- **Conventional Path Load** â€“ data is inserted to the database by using database SQL INSERT statements.
- **Direct Path Load** â€“ data is inserted into the database using data blocks that are written directly to the database. This method is much faster and therefore is the method selected for the data conversion process.


Copyright © 2026, Oracle and/or its affiliates.

SQL Loader has the ability to deal with multiple data files mapped to multiple tables as well as loading data into CLOB fields.

- Using SQL Loader When Processing Conversion Data Files

## Using SQL Loader When Processing Conversion Data Files

Processing of uploaded conversion data files is done via a Batch Process (for each staging table or MO) that does the following:

- Prepares parameters for the SQL Loader process and retrieves Control File for an input table or MO
- Invokes SQL Loader and processes the legacy data extract files that were uploaded to the dedicated File location on cloud
- Archive the input file(s) and upload the SQL Loader output files to the File location so they can be subsequently downloaded by the customer for review.

# Conversion Artifacts

The base product includes a set of artifacts that support the conversion process. These artifacts are provided on demand (generated as a result of a user request in the system).

The available artifacts are:

- Input Data Specifications
- SQL Loader Control Files
- Files List
- Conversion Input Data Specifications
- SQL Loader Control Files

## Conversion Input Data Specifications

The system will generate a text file with data specification instructions for each of the conversion tables that are supported by the base product.

The specification files are for reference ONLY and are not used by the system at run time. These files describe the data format for each table so that the data specified can be loaded correctly with the Control File that is also generated by the system.

For more details refer to Conversion Artifact Generator .

## SQL Loader Control Files

The control files are for internal use and are generated for each of the conversion tables that are supported by the product. These files are used by SQL Loader to interpret and load the data that is provided in the conversion data files.

The control file contains:

- General load parameters and options
- Data field specifications and table-specific load instructions

For more details refer to Conversion Artifact Generator .


Copyright © 2026, Oracle and/or its affiliates.

# Configuring Conversion Data Upload

The following sections describe the settings and configurations used by various conversion data upload processes.

- Conversion Master Configuration
- Conversion Task Types
- Conversion Artifact Generator
- Conversion Tasks
- Conversion Batch Controls

## Conversion Master Configuration

The **Conversion Data Upload Master Configuration** controls generic aspects of the legacy data load process.

The **Main** section captures the definitions globally applicable for all converted objects (tables and/or maintenance objects):

- **Default Conversion Instructions for Artifact Generator** reference conversion task types that define load parameters for individual tables or maintenance objects.
  The Master Configuration captures conversion task types with default configurations for:
  - Typical Table
  - Typical maintenance Object
  - Key Table.

**Special Instructions** section captures the definitions that support various idiosyncratic data upload scenarios

- **Override Conversion Eligibility â€" Table** list allows you to specify tables that are not marked for conversion in the metadata, but your implementation wishes to include in the conversion data upload and cleanup scope and vice versa - to exclude tables that are marked for conversion in the metadata from the data upload-related operations

> **Note**
>
> If the table you wish to convert has a related key table, this key table has to be explicitly added to this list.
>
> The sequences and the batch run history and other batch operation-related tables are never eligible for conversion operations.

- **Override Instructions** for **Tables** and **Maintenance Objects** define Conversion Instructions (conversion task types) for objects that require special processing or data formatting. For example, the default conversion instruction defines that CLOB data is supplied as a secondary file. The special instruction for a specific table may define that CLOB data is supplied in the main data file.


Copyright © 2026, Oracle and/or its affiliates.

# Conversion Task Types

To configure a new conversion task type, navigate using Admin > Conversion Support > Conversion Task Type.

Conversion task type controls the data load parameters for the target table or maintenance object. Conversion task type also defines the transactional business object for Conversion Task. The algorithms linked to the transactional business object are responsible for the generation of the Conversion Artifacts: control files, input data specifications and file lists.

> **Note**
>
> The product is delivered with several conversion task types whose names start with 'K1'. These configurations should not be changed directly as they reflect the default behavior of the product. The task types get refreshed with base values at each upgrade.. Use them as a templates; if needed, create a copy and modify it according to your requirements.

Conversion Task Types defines:

**General Input Data Instructions** (all Conversion Task Types Business Objects)

- **Control File Header** - a fragment of the control file that contains general load parameters. This fragment is included in the generated control file. This is a Foreign Key reference to the **Managed Content** entry.
- **Override Control File** - a full text of the control file that contains both load parameters and field definitions. The entire text is copied into generated control file 'as is'. This is a Foreign Key reference to the **Managed Content** entry.
- **Date and DateTime fields' formats**. These attributes are referencing the Extendable Lookup values for *K1-ConvLoadDateFormatLookup* and *K1-ConvLoadDateTimeFmtLookup business objects*
- Strings to use as **data delimiters** and **data enclosing characters**. These attributes are referencing the Extendable Lookup values for *K1-ConvFileDelimiterLookup* and *K1-ConvFileEnclosingCharLookup* business objects
- **CLOB data extract rule** indicates whether the CLOB data is supplied as part of the input data file or as a separate, secondary file
- **CLOB data line separator** defines how the records are separates in the secondary (CLOB) data file. This value should be set according to the operating system where the legacy data extract was created. This attribute is referencing the Extendable Lookup values for *K1-ConvCLOBLineSeparatorLookup* business object.

  For Linux CLOB Line Separator set the value to ***LF***, for Windows, set the value for ***CR+LF***

**CLOB Field Instructions** for Tables/MO-s that contain multiple CLOB fields (Business Objects *K1-ConvArtMultClobMOTaskType* and *K1-ConvArtMultClobTblTaskType*)

- Target Object - **Table** or **Maintenance Object**
- **Exclude CLOB Fields** list - the list of CLOB fields that are not expected to be provided with the legacy data extract and therefore should be excluded from the generated control file.

The following Conversion Task Types are pre-configured in the product:


Copyright © 2026, Oracle and/or its affiliates.

- **K1-CNV-TABLE** - meant to be used for a single table load. It is referencing a Transactional Conversion Task Business Object whose algorithms generate Table Conversion Artifacts
- **K1-CNV-MO** - meant to be used for a single maintenance object load. It is referencing a Transactional Conversion Task Business Object whose algorithms generate Maintenance Object Conversion Artifacts
- **K1-CNV-KEY-TABLE** - meant to be used for the Key Tables. Key Tables are index-organized tables (IOT) and as such cannot be loaded in multiple concurrent sessions of SQL loader. This Task Type is referencing a Control File Header (Managed Content) where ***Parallel Load*** parameter is set to ***false***

**Special configurations for tables that include more than one CLOB field**

- Configure a dedicated Conversion Task Type for tables with multiple CLOB fields. Determine and control which CLOB fields to migrate and avoid creating unnecessary empty data files.
- If you wish to migrate all CLOB fields, create a dedicated Conversion Task Type for the Table or Maintenance Object, using business objects *K1-ConvArtMultClobMOTaskType* or *K1-ConvArtMultClobTblTaskType*). Do not populate **Exclude CLOB Fields** list. As a result, the generated Control File and Input Data File Specifications will reference **all CLOB fields**.
- If you wish to migrate only some CLOB fields, create a dedicated Conversion Task Type for the Table or Maintenance Object, using business objects Business Objects *K1-ConvArtMultClobMOTaskType* or *K1-ConvArtMultClobTblTaskType*. Populate the **Exclude CLOB Fields** list. As a result, the generated Control File and Input Data File Specifications will **omit excluded CLOB fields.**
- If no dedicated Conversion Task Type is configured and defined on the Master Configuration as Override Instruction for the Table or MO, the system uses the default Conversion Instruction defined on the Master Configuration. In this case, the generated Control File and Input Data File Specifications will reference a **single CLOB field** among Table's multiple CLOB fields (the one with the lowest table field sequence).

# Conversion Artifact Generator

Conversion Artifacts is a set of supporting files used for data upload with SQL Loader. The artifacts are generated based on the table and maintenance object metadata and the definitions from Conversion Master Configuration and Configuration Task Types. Conversion artifacts include SQL Loader control files, input data file specifications and other supplemental files.

- **Input data file specifications** provide detailed field-by-field description of the expected data type and format, field order and other instructions. Your implementation may use these specifications to create the legacy data extract.
- **Control file** defines the rules and options of the data upload
- **File List** enumerates input data files expected to be provided for upload

The generator reads Conversion Master Configuration in order to determine the conversion task type associated with the table or maintenance object.

The generator creates **Conversion Tasks**, one instance per table and/or per maintenance object, creates attachments that store the artifacts and links the attachments to the Conversion Task as Related Objects.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> IMPORTANT: The regeneration is necessary each time the metadata or conversion configurations are amended.

Conversion Tasks query is provided with the base product.

You can choose to generate artifacts for table-level and/or maintenance object-level load, or for the entire system or for specific Table or Maintenance Object:

**Generate Artifacts in bulk â€" for all Tables and/or Maintenance Objects**

- Submit new batch job for Batch Control K1â€“CNVGA. This process performs a complete refresh of the conversion artifacts. It deletes existing and creates new Conversion Tasks for every Table and/or Maintenance Object that includes tables marked for conversion.

**Generate Artifacts for an individual Table or Maintenance Object**

- Navigate to **Admin**, then **Conversion**, then **Support**, then **Generate Conversion Artifacts**.
- Select one of the options:
  - **Table search** allows you to select specific target table
  - **Maintenance Object search** allows you to select specific target Maintenance Object.

> **Note**
>
> The process overrides all previously generated artifacts and replaces existing files.

**Accessing the Artifacts**

Navigate to Conversion Task query and search for the task that is corresponding to a table or a maintenance object. or use other available search criteria. Explore the Attachments collection to view the artifacts.

# Conversion Tasks

The following sections provide information about Conversion Tasks and generated conversion artifacts.

- Conversion Task Query
- Conversion Task

## Conversion Task Query

Conversion Task query allows to search by variety of criteria:

- Creating User, Conversion Task Type and Status
- Linked Object â€" Table or Maintenance Object
- Exact Conversion Task ID


Copyright © 2026, Oracle and/or its affiliates.

## Conversion Task

The purpose of the **Conversion Task** is to maintain the conversion artifacts for Table or Maintenance Object. Please consider the following with respect to conversion tasks.

- The business object lifecycle supports multiple attempts to re-generate the artifacts.
- The re-generation could be triggered either manually or by the Conversion Artifact Generator. Only one set of the artifacts is relevant at any point of time, hence only one Conversion Task at the time can exists for a given Table or Maintenance Object.
- Typically, Conversion Tasks are created by Conversion Artifact Generator. However, it can also be created manually by navigating to **Admin**, then **Conversion**, then **Support**, then **Conversion Task**.
- Use the Conversion Task Query to search for an existing entry. Select the entry from the search results to navigate to the maintenance portal to view and maintain the selected record.
- Conversion Task references a Conversion Task Type and contains standard business object record information, including the creation date/time, status update date/time and creating user.
- The task references the object - **Table or Maintenance Object** - for which the artifacts are generated.
- The collection of task's **Related Objects** store references to the Attachments that store a snapshot of the generated files. A new attachment is created after each generation run and linked to the Conversion Task for audit purposes.

# Conversion Batch Controls

The following batch controls are delivered with the base product:

| Batch Control | Description |
|---|---|
| K1-CNVLD | Conversion - Load Data using SQL Loader.<br>This process facilitates the legacy data extract upload using SQL Loader. The input data files are expected to be uploaded to a file storage location. See Input and Output File Location for more information about file storage locations.<br>One run uploads the data into a single target Table or Maintenance Object.<br>The process is multi-threaded.<br>This process is discussed in more details in the Data Load Batch chapter |


Copyright © 2026, Oracle and/or its affiliates.

| Batch Control | Description |
|---|---|
| K1-CNVDS | Conversion - Disable Conversion<br>This process resets the environment indicators and prevents further conversion activities. Once conversion is disabled, the following actions are blocked:<br>• Switching between Conversion and Production<br>• Table truncation<br>• Disable Indexes and Rebuilding the Statistics<br>• Data insertion into Production<br>• Data upload into staging<br>**Note**<br>The process should be executed only once and only after all of the conversion activities are completed and the application is ready for the real production use. |
| K1-CNVEN | Conversion - Enable Conversion<br>This process resets the environment indicators and enables it for conversion activities. It should be executed at the start of the Conversion project. |
| K1-DRPIN | Conversion - Disable Indexes on Production Tables<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the indexes on the batch run history and other batch operation-related tables can not be disabled<br>• If no specific target is populated, the process disables indexes on all converted tables in Production schema |
| K1-SDSIN | Conversion - Disable Indexes on Staging Tables<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the indexes on the batch run history and other batch operation-related tables can not be disabled<br>• If no specific target is populated, the process disables indexes on all converted tables in Staging schema |
| K1-DRICMS | Conversion - Disable Customer Modification Indexes<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object<br>• If no specific target is populated, the process disables indexes on all conversion-eligible tables in Customer Modification schema |
| K1-RIUSP | Conversion - Rebuild Indexes and Update Statistics in Production<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object<br>• If no specific target is populated, the process rebuilds indexes and updates statistics on all converted tables in Production schema<br>• Specify the sampling size (percentage) for statistics gathering. If not specified, the system will use default dbms_stats.auto_sample_size.<br>• Specify whether you wish to rebuild indexes, to gather table statistics or both.<br>It is recommended to rebuild the indexes and update statistics right after the data upload. |


Copyright © 2026, Oracle and/or its affiliates.

| Batch Control | Description |
|---|---|
| K1-RIUSS | Conversion - Rebuild Indexes and Update Statistics in Staging<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object<br>• If no specific target is populated, the process rebuilds indexes and updates statistics on all converted tables in Staging schema<br>• Specify the sampling size (percentage) for statistics gathering. If not specified, the system will use default dbms_stats.auto_sample_size<br>• Specify whether you wish to rebuild indexes, to gather table statistics or both<br>It is recommended to rebuild the indexes and update statistics right after the data upload. |
| K1-RUSCM | Conversion - Update Customer Modification Statistics<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object<br>• If no specific target is populated, the process rebuilds indexes and updates statistics on all conversion-eligible tables in Customer Modification schema<br>• Specify the sampling size (percentage) for statistics gathering. If not specified, the system will use default dbms_stats.auto_sample_size<br>• Specify whether you wish to rebuild indexes, to gather table statistics or both<br>It is recommended to rebuild the indexes and update statistics right after the data upload. |
| K1-CLNTB | Conversion - Truncate Tables in Production<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the sequences, the batch run history and other batch operation-related tables can not be truncated<br>• If no specific target is populated, the truncates all converted tables in Production schema |
| K1-SCLTB | Conversion - Truncate Tables in Staging<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object. Limitation: the sequences, the batch run history and other batch operation-related tables can not be truncated<br>• If no specific target is populated, the truncates all converted tables in Staging schema |
| K1-CLNCM | Conversion - Truncate Customer Modification Tables<br>This process can be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object<br>• If no specific target is populated, the truncates all conversion-eligible tables in Customer Modification schema |
| K1-CLNRT | Conversion â€” Truncate XML Resolution Tables<br>This process is allowed to run only in the Staging schema and may be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object<br>• If no specific target is populated, the truncates all XML Resolution tables in Staging schema |


Copyright © 2026, Oracle and/or its affiliates.

| Batch Control | Description |
|---|---|
| K1-CLNKY | Conversion - Truncate Key Reference Tables<br>This process is allowed to run only in the Staging schema and may be used in two modes:<br>• Populate Table or Maintenance Object batch parameter to process specific object<br>• If no specific target is populated, the truncates all Key Reference tables in Staging schema |

- Configuring Data Load Batch Controls for Conversion Project

## Configuring Data Load Batch Controls for Conversion Project

The generic batch process K1-CNVLD performs the load of the input data file(s) into a target table or maintenance object, specified as one of the batch parameters.

Consider creating individual batch controls per object (table or maintenance object).

It allows leverage the parallel processing capacity of the system; the data load processes for all objects can be submitted near-simultaneously, thus minimizing the time needed for the legacy data upload, which is only an initial step in the overall data conversion process.

Use batch control K1-CNVLD for rehearsal load batch run for various tables and/or maintenance objects and determine the optimal load strategy:

- Whether the data should be extracted and loaded as single Tables and/or Maintenance Objects
- Whether and how to split the data files and the number of threads for the batch
- Potential log file size and the feasible logging level

Configure data load Batch Controls for each individual Table or Maintenance Object. Pattern the configuration after K1-CNVLD and set the default parameter values for:

- **Input, Output and Archived File Storage Locations**. The location (compartments and buckets) has to be defined in advance in the cloud object storage; batch parameters are referencing Extendable Lookup ***F1-FileStorage*** value(s) corresponding to these locations. The parameter value should be composed as **file-storage:*//***<Extendable Lookup Value>/<object storage bucket name>, for example: *file-storage://OS-SHARED/CONV-Input*
- Target **Table** or **Maintenance Object** Either one of these two parameters must be populated.
- **File Extension**. When specifying the **file extension**, include all the extensions available. For example. if the file name is ***XXX.csv.gz,*** enter parameter value ***csv.gz***
- **Log Level.** Specify the value *LOG* to force SQL Loader to produce the most detailed data upload process log.
- **Retain Input Option**. This parameter controls whether the original input data files should be left 'as is' at the input data location or purged from the object storage or moved to the archived data location. When the input data is archived the timestamp is appended to the original file name.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Note: if the batch process was not able to upload the input data file, this file remains at the input data location.

- **Max Number of Upload Errors**.

  The value of this parameter pertains to the desired errors threshold for the SQL Loader. The load process halts after the specified number of records in error is reached. The SQL Loader log contains a message SKIPPED=NNN where NNN marks the position of the last record in error in the input data file. The process may then be resumed from this position if the batch is submitted with the ***skip*** parameter populated with the value of SKIPPED from the log.

Default **Thread Count** depends on the type of the uploaded table. Regular tables can be uploaded in a single or multiple threads. Key tables must be uploaded in a single thread

# Input Data Files

This section provides information on inputting data files for data conversion.


- Overview
- Input Data for a Staging Table
- Input Data for Maintenance Object
- Limitations for Input Data


## Overview

This section provides information on the requirements and details for file upload, loading table, maintenance object, CLOB, layout, data formatting, and so on related to inputting data files for data conversion.


- File Creation and Upload
- Loading Table or Maintenance Object
- Multiple Data Files
- Loading CLOB Data
- File Layout and Data Formatting
- Defining Custom Data Delimiters and Enclosing Characters


### File Creation and Upload

The extract data may be uploaded in both compressed and decompressed form.

Supported compressed formats include zip and gzip. The files with extension **.gz** and **.zip** will be automatically decompressed by the system.

> **Note**
>
> For the further details on supported compressed formats please refer to the Oracle Utilities Application Framework documentation.



Copyright © 2026, Oracle and/or its affiliates.

**File Creation**

Use Conversion Task query to retrieve conversion task(s) for table(s) or maintenance object(s) that you are planning to convert. Review input data specifications. Create the data extract text file, using field delimiters, enclosing characters and data formatting according to the specifications.

If the files were created under Windows operating system, it is recommended to remove the carriage return character prior to compression (not applicable for secondary files with CLOB data).

**File Compression with gzip**

Use gzip utility to compress the file.

- For Windows operating system, download and install gzip utility for Windows. Run the following:

```
# Navigate to the gzip folder
cd C:\gzip-1.3.12-1-bin\bin
# Run gzip.
# New file will be created in the same directory as the original the input file
gzip -k <full path to the input data file>
# Sample command for the file TABLE_NAME.txt located under
# C/myDataDirectory/:
gzip â€“k C/myDataDirectory/TABLE_NAME.txt
```

- For Unix/Linux operating systems, run the following command:

```
gzip < <inputFile name> > <outputFile name with .gz suffix>
# Sample command for the file TABLE_NAME.txt
gzip < TABLE_NAME.txt > TABLE_NAME.txt.gz
```

## Loading Table or Maintenance Object

The conversion data upload supports two types of the extract:

- Data extract that contains the data for a single target staging table.

  In this case, the upload process inserts records into a single staging table. The input data file consists of similar records, each represents a row in the target table.

- Data extract that contains the data for the group of target staging tables that represent a Maintenance Object.

  In this case, the upload process populates all the tables that belong to a maintenance object. The input data file contains multiple types of records that are inserted into various tables.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> If the data is uploaded for a Maintenance Object, each record has to be prefixed with the table name that is used as record type indicator. This requirement is also mentioned in the input file specification

- The application stores the large volumes of data in partitioned tables. If the Maintenance Object contains partitioned tables, the data should be uploaded in a certain order:
  - During the data upload, the primary table in the Maintenance Object should be loaded first, then the indexes should be rebuilt, followed by the data upload into the child tables

## Multiple Data Files

The large volume of the converted data may require the extract to be split into multiple parts.

In this case the multi-part extract for a target staging Table or Maintenance Object is loaded with a multi-threaded batch run.

**Sample scenario**:

10M of the Person records are being converted. The extract file is split into 10 files, PERSON1, PERSON2 PERSON3 etc., each file contains 1M records. All 10 files are uploaded to the input file location.

Batch process 'Load Person data' is submitted with 5 threads; each thread picks up 2 files and uploads the data into staging tables.

## Loading CLOB Data

The CLOB data can be loaded in two ways:

- Extracted as part of the main data file. This method is suitable for the relatively small CLOB data size. Refer to Limitations for more details.
- Extracted into a separate file. This method is suitable for larger CLOB data sizes and also for large volume tables or maintenance objects. In this scenario, the secondary file has to be uploaded at the same time as the primary data file.

> **Note**
>
> When loading CLOB data as a secondary file, a separate file is created for each CLOB field in the. It means that if the table 'XXX' contains CLOB fields 'clob1Field1' and 'clobField2', the system would expect to process files XXX, XXX_clobField1_CLOB and XXX_clobField2_CLOB. See also Notes on Secondary Data Files

- Notes on Secondary Data Files (CLOB Data)

## Notes on Secondary Data Files (CLOB Data)

Secondary data file is loaded simultaneously with the primary data file. The CLOB data extract should contain exactly the same number of records as a primary file.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> When splitting the extract into multiple data files, make sure that the numbers of extracted records in the primary and secondary files are matching.
>
> It is applicable for both table-level and maintenance-object level load.

**Sample Scenario:**

Table CI_PER includes CLOB column C1_PER_DATA_AREA. The CLOB field is used to store customer's resume.

The legacy table contain 10K records and the CLOB data is available for some of the records

The primary data file, CI_PREM.csv contains 10K lines, one line per record

*The secondary data file, CI_PREM_CLOB, contains 10K records too. The records with the empty CLOB are represented by CLOB delimiter.*

| Primary File – Regular Data Fields | Secondary File - CLOB |
|---|---|
| 12345678,John,Doe,..... | John's resume <endclob> |
| 45165609,Smith,Mary,..... | Mary Smith resume<endclob> |
| 33336733,Brand,Brent,..... | <endclob> |
| 06737482,Sunshine,Julie,..... | Julie Sunshine resume<endclob> |

**Splitting single table data extract with CLOB**

The following example illustrates how to split the single table large volume extract correctly if the CLOB data is supplied in the secondary file:

Assuming the table has 10K records, split into two equal parts. The following files should be created:

- Primary File 1 with 5K records
- Secondary File 1 with 5K CLOB entries
- Primary File 2 with 5K records
- Secondary File 2 with 5K CLOB entries

**Splitting maintenance object data extract with CLOB**

The following example illustrates how to split the maintenance object large volume extract correctly if the CLOB data is supplied in the secondary file:

Assuming the maintenance object has two tables, one of the tables includes CLOB column and the primary table has 10K records. The data supposed to be split into two equal parts. The following files should be created:

- Primary File 1 with 5K records for the primary table and corresponding records for the child table
- Secondary File 1 with CLOB entries corresponding to the table in the Primary File 1
- Primary File 2 with 5K records for the primary table and corresponding records for the child table


Copyright © 2026, Oracle and/or its affiliates.

- Secondary File 2 with CLOB entries corresponding to the table in the Primary File 2

> **Note**
>
> For Maintenance Object-level load consider supplying CLOB data in the primary file, to avoid complicated logic of data splitting.
>
> Refer to CLOB Data - XML for limitations concerning XML CLOB data

## File Layout and Data Formatting

The input data specifications file for each Table or Maintenance Object that are subject to conversion are stored as an attachment linked to the corresponding conversion tasks. The Conversion Artifact Generator creates conversion task for each table and/or maintenance object.

Specifications are generated based on the metadata and the conversion configurations defined on Conversion Task Type and Conversion Master Configuration. Configurations control delimiters and enclosing characters and define CLOB data processing rules.

The data extract file is expected to contain one record per each target table row. The data field order in the record has to be exactly the same as it is listed in the specification file.

The specifications contain:

- Data delimiters and expected secondary data file names
- Field-by-field data formatting rules for

  data types and size, Date, DateTime and Time formatting
- Special notes on CLOB fields

The fields in the record are expected to be separated by a delimiter character (or string).

For the not-nullable columns with no data, a single space enclosed in the enclosing character (or string) has to be supplied, for example " "

The CLOB data in the main record has to be enclosed in the CLOB Delimiter string, for example *<clobend>some clob data<clobend>*

The CLOB records in the secondary data file has to be separated by CLOB Delimiter string followed by the new line sign, defined as the CLOB Line Separator. IMPORTANT! the last entry in the CLOB data file has to end with the new line, too

- If the data extract is created on Linux, use Line Feed to separate CLOB records and make sure that there is no space after CLOB Delimiter string
- If the data extract is created on Windows, use Line Feed followed by Carriage Return.

Enclosing character (or string) can also be used to encapsulate data that includes a field delimiter, for example *"Hello, World"*

## Defining Custom Data Delimiters and Enclosing Characters

The default configurations are delivered with the base product.


Copyright © 2026, Oracle and/or its affiliates.

Your entire legacy data or a specific table, or the data extract mechanism may require use of a different data delimiter and/or enclosing characters. To enable alternative delimiters, perform the following steps:

- Define new data delimiter string, enclosing string, CLOB delimiter string and/or CLOB Line Separator using the corresponding Extendable Lookups
- Configure Conversion Task Type using newly created delimiters
- Amend Conversion Master Configuration, setting default (for all tables or maintenance objects) or override (for specific table or maintenance Object) instructions.
- Regenerate Conversion Artifacts

Now you can create data extract according to the newly generated specifications.

# Input Data for a Staging Table

The extract for a single table contains delimited data records. Depending on the conversion configuration, the CLOB data should be supplied either inline or in a separate, secondary file.

**Input Data File Names**

Input data file names has to follow the rules described in this chapter. The names are case-sensitive.

| File Type | File Name |
|---|---|
| Single data file per table, no CLOB | File name equal target table name, such as CI_PREM |
| Single data file per table, **single** CLOB column in the table, CLOB data in secondary file | Primary file name equal target table name, such as CI_PREM<br>Secondary file name is composed as target table name concatenated with '_CLOB', such as CI_PREM_CLOB |
| Single data file per table, **multiple** CLOB columns in the table, CLOB data in secondary file | Primary file name equal target table name, such as CI_PREM<br>**For each CLOB column** being uploaded, the secondary file name is composed as target table name concatenated with CLOB column name concatenated with '_CLOB', such as CI_PREM_COL1_CLOB, CI_PREM_COL2_CLOB |
| Multiple data files per table, no CLOB | File name is composed as target table name concatenated with file number, such as CI_PREM1, CI_PREM2 |
| Multiple data files per table, CLOB data in secondary file | Primary file name is composed as target table name concatenated with numeric file number, such as CI_PREM1, CI_PREM2, and so on.<br>Secondary file name is composed as target table name concatenated with '_CLOB' and file number, such as CI_PREM_CLOB1, CI_PREM_CLOB2, and so on. |


Copyright © 2026, Oracle and/or its affiliates.

| File Type | File Name |
|---|---|
| Multiple data files per table, **multiple** CLOB columns in the table, CLOB data in secondary file | Primary file name is composed as target table name concatenated with numeric file number, such as CI_PREM1, CI_PREM2, and so on.<br>**For each CLOB column** being uploaded, the secondary file name is composed as target table name concatenated with CLOB column name concatenated with '_CLOB' and file number, such as CI_PREM_COL1_CLOB1, CI_PREM_COL1_CLOB2,<br>CI_PREM_COL2_CLOB1, CI_PREM_COL2_CLOB2, and so on. |

# Input Data for Maintenance Object

The extract for a maintenance object contains delimited data records for all tables included in the maintenance object. Depending on the conversion configuration, the CLOB data should be supplied either inline or in a separate, secondary files, one file per table containing CLOB. Refer to Notes on Secondary Data Files (CLOB Data) for limitations concerning XML CLOB data and more details.

### Input Data File Names

Input data file names has to follow the rules described in this chapter. The names are case-sensitive.

File names for Maintenance Object contain Maintenance Object Code. If the code includes spaces or special characters, it should be replaced with '_', for example:

*For actual MO Code: SP/MTR HIST, in the file name, use SP_MTR_HST*

| File Type | File Name |
|---|---|
| Single data file per Maintenance Object, no CLOB | File name equal target maintenance object name, such as PREMISE |
| Single data file per Maintenance Object, **single** CLOB column in the table, CLOB data in secondary file | Primary file name equal maintenance object name, such as PREMISE<br>Secondary files name composed as maintenance object name concatenated with table name concatenated with '_CLOB'.<br>For example, if PREMISE's CI_PREM table contains CLOB, the secondary file name is: PREMISE_CI_PREM_CLOB |
| Single data file per Maintenance Object, **multiple** CLOB columns in the table, CLOB data in secondary file | Primary file name equal maintenance object name, such as PREMISE<br>**For each CLOB column** being uploaded, the secondary files name composed as maintenance object name concatenated with table name concatenated with CLOB column name concatenated with '_CLOB'.<br>For example, if PREMISE's CI_PREM table contains two CLOB columns COL1 and COL2 , the secondary file names are: PREMISE_CI_PREM_COL1_CLOB, PREMISE_CI_PREM_COL2_CLOB |


Copyright © 2026, Oracle and/or its affiliates.

| File Type | File Name |
|---|---|
| Multiple data file per Maintenance Object, no CLOB | File name is composed as target maintenance object name concatenated with file number, such as PREMISE1, PREMISE 2 etc |
| Multiple data file per Maintenance Object, single CLOB column in the table, CLOB data in secondary file | Primary file name is composed as target maintenance object name concatenated with file number, such as PREMISE1, PREMISE 2 etc<br>Secondary files name composed as maintenance object name concatenated with table name concatenated with '_CLOB' concatenated with file number.<br>For example, if PREMISE's CI_PREM table contains CLOB, the secondary file names are: PREMISE_CI_PREM_CLOB1, PREMISE_CI_PREM_CLOB2, etc |
| Multiple data file per Maintenance Object, **multiple** CLOB columns in the table, CLOB data in secondary file | Primary file name is composed as target maintenance object name concatenated with file number, such as PREMISE1, PREMISE 2 etc<br>**For each CLOB column** being uploaded, the secondary files name composed as maintenance object name concatenated with table name concatenated with column name concatenated with '_CLOB' concatenated with file number.<br>For example, if PREMISE's CI_PREM table contains two CLOB columns COL1 and COL2 , the secondary file names are: PREMISE_CI_PREM_COL1_CLOB1, PREMISE_CI_PREM_COL1_CLOB2, PREMISE_CI_PREM_COL2_CLOB1, PREMISE_CI_PREM_COL2_CLOB2 etc |

# Limitations for Input Data

### Key Tables

Due to technical limitations, Key tables cannot be loaded in multi-threaded mode. If you decide to create input data files using the Maintenance Object method, consider the following:

- Create data extract(s) for a Maintenance Object data, but exclude the Key table(s) and run data upload in multi-threaded batches
- Create separate extract for the Key table(s) only and upload them in single threaded batches.

### Duplicate Records

Duplicate records are not rejected during the upload. This may result in the index corruption and other issues in the target table(a).

The default SQL Loader data upload method configured in the system is Direct Path. Among the advantages of a Direct Path method are higher performance and more efficient resources utilization. Constraints are disabled during Direct Path load and as a result the duplicate PK records are not rejected.

There are two approaches to address the issue

- Clean up the duplicate records from the legacy data extract BEFORE the upload


Copyright © 2026, Oracle and/or its affiliates.

- Customise the default configuration and switch to the Conventional Path load method.
  - **Create custom Control File Header**: Duplicate Managed Content K1-SQLLoaderControlFileHeader .and modify to use Conventional Path Load (Consult SQL Loader documentation for more detailed information).
  - **Modify Conversion Instructions** (Conversion Task Type): specify SQL Loader Control File Header created above
  - **Regenerate Conversion Artifacts**

**CLOB Data Size**

The maximum size of the CLOB data included in the main data file is 10K. If converted CLOB data size is expected to be larger than 10K, create secondary data files.

**CLOB Data - XML**

The XML data formatted using new line characters cannot be uploaded as part of the primary data file. If your CLOB data includes formatted XML, you should create secondary data file(s) for CLOB columns.

# Conversion Orchestration

This section provides information on orchestrating a data conversion.

- Input and Output File Location
- Understanding Load Data Process
- Understanding the Conversion Orchestration Process

## Input and Output File Location

The following locations has to be defined in order to support legacy data upload:

- File location for input data files upload
- File location for SQL Loader output files and also for archived input data files

See Object Storage chapter for more details on how to define the file location.

## Understanding Load Data Process

The setup of the batch controls that is necessary to support data conversion and legacy data extract upload is described in Data Load Batch Controls .

The following sections explain the process of legacy data extract upload with SQL loader.

- Load Data Batch Run Parameters
- Multi-Threading
- Load Outcome â€“ Log, Bad and Rejected Files
- Interrupted Load
- Tables Cleanup and Index Maintenance
- Load Performance Tuning


Copyright © 2026, Oracle and/or its affiliates.

## Load Data Batch Run Parameters

The data upload batch is processing single target object â€“ Table or Maintenance Object. The data extract may be supplied in one or multiple files, created and named according to the Input File instructions for a Staging Table and Maintenance Object .

The process is multi-threaded and optimized for the parallel load.

Using the batch parameters you can control:

- Level of logging
- Data extract retention
- Resume the interrupted load
- Maximum number of load errors

## Multi-Threading

The process is threading by dividing the multiple data files between the threads. Each thread in turn, is processing its files one by one.

The exception is Key Tables. They are index-organized tables and have to be loaded in the single thread, due to the SQL Loader restrictions.

## Load Outcome â€“ Log, Bad and Rejected Files

SQL Loader output files are available for download after each run . Refer to Input and Output File Location for details. To ensure file name uniqueness, the output file names are concatenated with the current timestamp.

- **Log file** contains the information about the run, including number of record processed, errors encountered and other useful information. The level of logging is controlled by the batch parameter ***log*** .
- **Bad** and **Rejected** files contain the records that were rejected by either the SQL Loader or the database. If the errors are not extensive and could be fixed manually, these files can be edited, renamed and used as an input again.

## Interrupted Load

The data load process can be re-started. Once SQL Loader reached the maximum number of errors allowed for the run, it records the position of the last record in error in the log. The restarted load begins pulling the data from the file after skipping certain number of records.

In order to continue processing interrupted load, perform the following steps:

- Retrieve the log file. Find statement '*Specify SKIP=* ***n*** *when continuing the load .'* in the log file.
- Examine the log file and determine whether the problems with the data are easily amendable. Retrieve .bad and .rejected files from the same location where SQL Loader log files are found, review the problem and decide whether the entire extract has to be corrected or the upload should continue forward.
- Re-upload the input data file to the server
- Submit the batch once again, this time specifying the ***n*** as a ***skip*** batch parameter


Copyright © 2026, Oracle and/or its affiliates.

Simillar approach can be utilised to recover from the partial data upload. Note that in order to provide an additional safeguard for your effort and allow recovery it is recommended to create the data extract with records ordered by the primary key. This technique will allow you to determine the portion of the remainder of the un-uploaded data and make decision on the subsequent actions. If you are uploading large volumes of data and the interruption happened toward the end of the upload process it could be beneficial to complete the upload rather than to cleanup and start over.

Consider the following scenarios:

- Data Upload batch process ended abnormally due to an unrecoverable event such as software malfunction. In this case neither SQL loader nor the batch process are able to end gracefully and report the number of records processed.
- Data Upload batch process encountered a legitimate error and the batch thread and the entire batch run is now in error status.
- The SQL Loader process ended with fatal error that prevented it to log the "skipped" option properly and it is not found anywherein the SQL Loader log

If you decied to proceed with recovery, perform the following steps:

- Query the table and determine the primary key of the last record that was sucessfully uploaded - this would be possible if the data in the input file was ordered
- Then determine the position of this record in the data file
- Finaly, submit the batch once again and use the ***skip*** batch parameter to resume the load from this position

## Tables Cleanup and Index Maintenance

The SQL Loader runs in parallel mode in order to maximize performance. Therefore the target tables are not emptied prior to the data insertion and must be done explicitly, by running corresponding batch jobs.

Indexes are disabled during the data upload and then need to be re-built afterwards. The product provides batch processes that support index disabling and re-building indexes and gathering table statistics.

Most of the larger volume tables in the application are partitioned. The partitioned indexes cannot be disabled implicitly by the data upload process and should be disabled implicitly before running the upload.

If your are migrating an existing on-premise Oracle Enterprise Utilities customer to the cloud, the migration requirements may include the data in the custom tables. On cloud, these tables reside in the Customer Modification schema. The product provides the dedicated batch controls to operate on the Customer Modification schema.

- Use batch control ***K1-SCLTB*** to truncate tables in **Staging** area and batch control ***K1-CLNTB*** to truncate tables in **Production**. Use batch control ***K1-CLNCM*** to truncate tables in Customer Modification schema.
- Use batch control ***K1-SDSIN*** to disable indexes in **Staging** area and batch control ***K1-DRPIN*** to disable indexes in **Production**. Use batch control ***K1-DRICM*** to disable indexes on tables in the Customer Modification schema.
- Use batch control ***K1-RIUSS*** to rebuild indexes and/or update table statistics in Staging and ***K1-RIUSP*** to perform the same in **Production**. Use batch control ***K1-RUSCM*** rebuild indexes and/or update statistics on tables Customer Modification schema.


Copyright © 2026, Oracle and/or its affiliates.

- Use batch control ***K1-CLNKY*** to truncate **Key Reference** tables in **Staging** area
- Use batch control ***K1-CLNRT*** to truncate **XML Resolution** tables in Staging Area
- Use batch process ***K1-SDSTG*** to disable Triggers in **Staging** and batch control ***K1-PDSTG*** to disable triggers in **Production**
- Use batch process ***K1-SENTG*** to enable Triggers in **Staging** and batch control ***K1-PENTG*** to enable triggers in **Production**

Supported scenarios include:

- **Truncate Specific Table or Maintenance Object**: In order to truncate the specific table, submit the corresponding batch job, specifying the `table` parameter. In order to truncate the data stored in a specific Maintenance Object, submit the corresponding batch job, specifying the `maintenanceObject` parameter. The process will also disable indexes in the truncated tables.

  > **Note**
  >
  > If the truncated table's metadata defines a Key table, the Key table gets truncated too.

- **Truncate all Converted Tables**: In order to truncate all converted tables run the corresponding batch job without specifying either the `table` or `maintenanceObject` parameters. The process will disable indexes and truncate all tables marked for conversion in the metadata.
- **Truncate Key Reference Tables or XML Resolution Tables for Specific Table or Maintenance Object**: In order to truncate Key Reference table linked to a staging table, submit the corresponding batch job, specifying the `table` parameter. In order to truncate Key Reference tables for a specific Maintenance Object, submit the corresponding batch job, specifying the `maintenanceObject` parameter.
- **Truncate all Key Reference Tables or XML Resolution Tables**: Submit the corresponding batch job without specifying either the `table` or `maintenanceObject` parameters to truncate all Key Reference tables.
- **Disable Indexes for Specific Table or Maintenance Object**: In order to disable indexes for the specific table, submit the corresponding batch job, specifying the `table` parameter. In order to disable indexes for the specific Maintenance Object's tables, submit the corresponding batch job, specifying the `maintenanceObject` parameter.
- **Disable Indexes for all converted Tables**: In order to disable indexes on all converted tables run the corresponding batch job without specifying either the `table` or `maintenanceObject` parameters. The process will disable indexes and truncate all tables marked for conversion in the metadata.

> **Note**
>
> You should not use conversion batch processes to truncate and/or alter indexes on the batch and batch job stream-related tables. Attempt to cleanup the batch job history or reset sequences may render the system un-operational.


Copyright © 2026, Oracle and/or its affiliates.

## Load Performance Tuning

The data specifics â€“ volumes, structure and other aspects â€“ may justify customizing the default configurations for all or some tables.

The SQL Loader is processing data according to the Control File. The fragment that contains general load parameters is stored as Managed Content that is referenced on the Conversion Task Type.

You can create new Managed Content, configure the Conversion Task Type and specify the new settings on Conversion Master Configuration, for all or specific Tables or Maintenance Objects.

# Understanding the Conversion Orchestration Process

The following sections provide information on general aspects of conversion orchestration.

**Enabling Conversion**

Conversion activities are possible as long as conversion is enabled in the environment. The authorized Conversion Administrator may run the corresponding batch job to enable conversion.

**Switching between Staging and Production Schemas**

In the environment enabled for conversion, the online application is running against the database schema that contains synonyms pointing to either Staging or Production tables.

The switching is possible only while Conversion is Enabled

Navigate to **Admin > Conversion Support > Switch Schema** to switch the synonyms, .

The current schema information is displayed on the sidebar zone when Conversion is enabled in the environment.

> **Note**
>
> The operation fails if there are batch processes currently in progress. Should you receive the error message while attempting to switch synonyms, please review the running batches and wait until its all completed before repeating the attempt.

**Preparing for Conversion**

The preparation steps include evaluating initial conversion configurations provided by the product, customizing the configurations if needed, and finally, generating conversion artifacts.

Most of the preparation should be done when the environment is running against the Production schema, except for job streams for conversion processes that are scheduled and, published when the schema is switched to Staging.

**Conversion Development**

The creation of the legacy data extract and rehearsal of the data load and subsequent conversion data processing is done while the environment is running against the Staging schema.


Copyright © 2026, Oracle and/or its affiliates.

Consider creating and scheduling the job stream for the sequence of the batch processes for both data load and following validation, key generation and insertion.

**Uploading Data**

SQL Loader implicitly disables the indexes in the non-partitioned tables before the upload. In the partitioned tables the indexes needs to be explicitly disabled by the batch process prior to the upload. Once the data upload is completed the indexes have to be rebuilt.

Upload has to be done in a certain order if loading single tables that belong to a Maintenance Object whose primary table is partitioned. The child tables within such Maintenance Objects contain Foreign Key constraints to the primary table's Primary Key. The primary table has to be uploaded first. The indexes should be rebuilt right after upload. The subsequent tables should be uploaded according to the Maintenance Object table's hierarchy (parent first, child next)

You may combine the table truncation, data upload and the index maintenance and statistic updates into a batch stream and create a dedicated batch job stream for each converted Table or Maintenance Object.

**Conversion Run**

For the mock-up, or dress rehearsal or the actual go-live conversion run you may choose to amend data load batch controls for tables/maintenance objects to improve performance:

- Reduce the logging level - set ***log*** batch parameter to NOLOG
- Do not retain input data - set ***retaininput*** batch parameter to PURGE

**Disabling Conversion**

After conversion is completed, the various activities such as table truncation and index maintenance should not be allowed.

Upon conversion project completion, switch the schema to Production and request that the authorized Conversion Administrator run the corresponding batch job to disable Conversion in the environment.

**Enabling Incremental Conversion**

Legacy data conversion may be done in stages. After the first chunk is converted and the system begins running in production, you may need to convert the next portion(s) of the data. This orchestration macro-scenario is called *incremental conversion*; it imposes certain limits and restrictions on the conversion activities and it has to be approved by the management.

If the incremental conversion has been approved, request that the authorized Conversion Administrator run the corresponding batch job to enable the incremental conversion in the environment.

# Security Setup for Conversion

The services associated with data conversion are combined into the following user groups:

- **Conversion Administrators** includes the services linked to the most sensitive database operations: enable/disable conversion and switching database synonyms between production and staging schema
- **Conversion Development** includes the services associated with conversion-related configurations


Copyright © 2026, Oracle and/or its affiliates.

- **Conversion Operations** provides access to services associated with data load batch processes. This group could be extended to include services associated with specific table(s) or maintenance object data loads

These user groups are included in the base product. Your implementation may chose to use it as is or create a different setup.

## Conversion Reconciliation Reports

Your cloud service subscription includes online database querying capabilities via Analytics Publisher and Database Actions. These can be used to create and run data reports against both production and staging schema tables. Contact your Security Administrator for the access information.

# Batch Scheduler for Cloud Implementations

This chapter provides information and requirements on batch scheduler in support of Oracle Utilities Cloud implementations.


- SaaS Batch Scheduler Overview
- Scheduler Batch Stream Definitions
- Batch Scheduler Batch Control
- Scheduler Batch Stream Operations


## SaaS Batch Scheduler Overview

SaaS Batch Scheduler is a feature in Oracle Utilities Cloud Service Foundation that provides a SaaS batch scheduler engine and user interfaces. It facilitates customers and implementers to define, manage and schedule stream of batch jobs to run at periodic intervals using a user friendly interface. The SaaS batch scheduler engine is responsible for scheduling and processing the stream of batch jobs. It has a processing loop that runs every X seconds(configured by default at 10 seconds) to perform these operations. In addition, users can monitor the progress of the streams, each of the steps and also perform required exception handling. The SaaS batch scheduler engine processes the stream of batch jobs per scheduled time, within that minute.

SaaS Batch Scheduler feature is supported by the following system entities:

- Scheduler Batch Stream Definition â€“ a Business Object based on the Batch Job Stream Definition maintenance object
- Batch Scheduler Batch Control â€“ a Business Object based on the Scheduler Program maintenance object

## Scheduler Batch Stream Definitions

This section provides information on Scheduler Batch Stream Definitions.


- Defining a Scheduler Batch Stream Definition
- Activating a Scheduler Batch Stream Definition




Copyright © 2026, Oracle and/or its affiliates.

- Updating an Active Scheduler Batch Stream Definition
- Inactivating a Scheduler Batch Stream Definition
- Canceling a Scheduler Batch Stream Definition
- Deleting a Scheduler Batch Stream Definition
- Migrating Scheduler Batch Stream Definitions
- Tips For Setting Up Scheduler Batch Stream Definitions
- System Override Date Support

# Defining a Scheduler Batch Stream Definition

In order to define a Scheduler Batch Stream Definition, from the **Admin** menu, select **Batch Operations**, select **Scheduler Batch Stream Definition**, and select **Add**.

A Scheduler Batch Stream Definition includes specifying the batch jobs and the order in which they should be run as "Stream Steps", the user to use to run these batch jobs and a "Schedule" to state when and how often the batch job stream should run. Batch jobs can be setup to run in sequence or in parallel fashion or a combination of both.

**Business Date**: A Scheduler Batch Stream Definition can be optionally configured to run on a particular business date. When a business date is specified, each of the batch jobs in the stream run with the business date set. The business date can be either a specific business date or a variable business date. For example, business date set to run a day before current date. This ensures every time the stream runs the batch jobs in the stream run with current date -1.

**Run On Activate**: A Scheduler Batch Stream Definition can also be optionally configured to run as soon as it's activated.

**Start Schedule Date/Time**: It can optionally be configured to specify schedule to be effective in the future by setting a date/time to start schedule. If the start schedule date/time is not provided then the schedule on the Scheduler Batch Stream is effective as of current date.

The batch job to run for each stream step can be either specified directly or via a "Batch Scheduler Batch Control". A Batch Scheduler Batch Control enables setting custom batch parameters on a Batch Control. When a batch job is submitted for a stream step that has Batch Scheduler Batch Control specified on it, the custom batch parameters are used on the submitted batch job. Refer to Configuring a New Batch Scheduler Batch Control for more details.

A Stream step includes specifying the following details

- Step Name: the name of the step
- Batch Control: the batch control to run for that step.
- Thread Count: the number of threads to use to run this step's batch control
- Batch Scheduler Batch Control: the batch scheduler batch control to use for that step. Either a batch control or batch scheduler batch control can be specified but not both
- Step Conditions: specifies when the current step should run. It includes the following details
  - Step Name: the step that should have run prior to current step
  - Step Status: the status the prior step should be in when its finished


Copyright © 2026, Oracle and/or its affiliates.

– Step Condition: indicates how (And/Or) prior step should combine with next subsequent step conditions. For example, if step C should execute after step B and A have ended, then step C's has two step conditions 1)A Ended And 2)B Ended.

> **Note**
>
> There can be stream configurations where a step is specified and no steps run after the specified step runs. This is not a common scenario and a warning is logged on the Log tab to adjust stream configuration, if needed.

A Scheduler Batch Stream Definition's schedule can be specified using various frequencies.

- Daily, By Minutes: run every "x" minutes every day except on specified excluded days
- Daily, By Time: run at certain times every day except on specified excluded days
- Monthly, By Month Day Number: run on a specific day of the month, at specified time
- Monthly, By Week Days: run on days of a specific week in a month, at specified time
- Weekly: run every "x" weeks on specific days, at specified time
- Yearly, By Month Day Number: run on a day of the month, at given time in specified months
- Yearly, By Week Days: run on days of a specific week, at given time in specified months
- Custom: run on a specific date at specified time

A Scheduler Batch Stream Definition can be setup to have level of service algorithms. Following three level of service algorithm types are supported.

- Stream Run has failed.
- Scheduler Batch Stream not started X minutes.
- Scheduler Batch Stream Job Ran Too long (Relative Run).

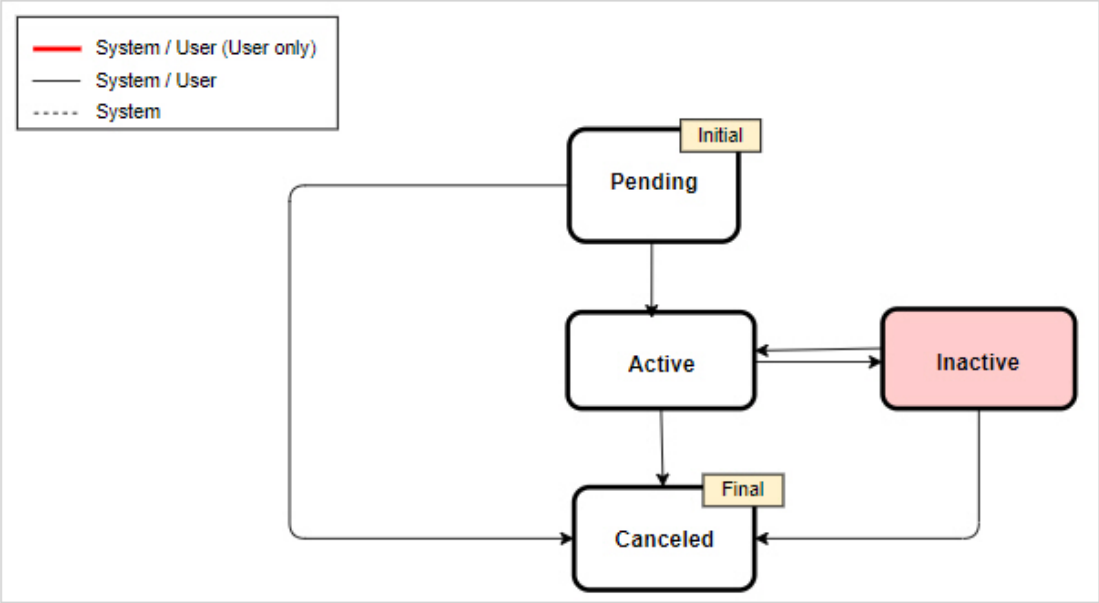
A Scheduler Batch Stream Definition can also be created without defining a schedule. When activated, it remains activated without a schedule and can be used for any manual runs.

There are custom ways in which a scheduler batch stream definition can be setup. This involves creating a Batch Scheduler Batch Control. Refer to Tips For Setting Up Batch Job Stream Definitions and Batch Scheduler Batch Control for more information.

## Activating a Scheduler Batch Stream Definition

Once the Scheduler Batch Stream Definition is ready, it can be activated in the SaaS batch Scheduler to run at periodic intervals by transitioning the Scheduler Batch Stream Definition to Active state. An activated Scheduler Batch Stream Definition is called a Scheduler Batch Stream. A Scheduler Batch Stream runs based on the schedule defined on it.


Copyright © 2026, Oracle and/or its affiliates.




By default, every batch job in a Scheduler Batch Stream Definition is internally setup to run in such a way that if that batch job fails the Scheduler Batch Stream run stops execution. User then has an option to fix the issue and re-start the stream.

## Updating an Active Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be changed while in Active state. While active, any changes made to the Scheduler Batch Stream Definition are reflected on the next Scheduler Batch Stream run.

Changes to a Scheduler Batch Stream Definition can be made by editing it or using the edit options in the Scheduler Batch Stream Definition steps tab.

## Inactivating a Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be stopped from running further by inactivating it. Once inactivated, the SaaS batch scheduler stops scheduling and processing the scheduler batch stream definition. This is useful in situations where the stream needs to stop running temporarily for a brief period of time.

A Scheduler Batch Stream Definition can be re-activated back for scheduling and running in the future.

## Canceling a Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be canceled so that it no longer runs. If a scheduler batch stream run is ongoing when the scheduler batch stream definition is canceled, a warning is used. Canceling a scheduler batch stream definition has no impact on that currently ongoing run. User has the option to continue with the cancelation of the scheduler batch stream definition and let the scheduler batch stream run or navigate to the scheduler batch streams operations portal to cancel the currently ongoing run and then cancel the scheduler batch stream definition. Canceled scheduler batch stream definitions are allowed to be deleted.


Copyright © 2026, Oracle and/or its affiliates.

# Deleting a Scheduler Batch Stream Definition

A Scheduler Batch Stream Definition can be deleted in Pending, Inactive and Canceled states. In case of deleting an Inactive or a Canceled scheduler batch stream definition, respective scheduler batch stream runs are also deleted.

# Migrating Scheduler Batch Stream Definitions

Scheduler Batch Stream Definitions can be migrated to other environments using the Content Migration Assistant (CMA) process or the Process Automation Tool.

Scheduler Batch Stream Definitions and Batch Scheduler Batch Controls are added in the target environment in the same status as the source environment. That is active scheduler batch stream definitions are added as active.

A Scheduler Batch Stream Definition should be marked as Inactive in source environment before migrating to target environment if it should not run as soon as it's migrated.

> **Note**
>
> If a Scheduler Batch Stream is already scheduled in target environment and has next scheduled run date/time populated and the migration from source environment results in the stream having a different schedule, the new schedule will not be picked up by SaaS batch scheduler until the next run happens on the previously scheduled time. In order for the new schedule to take effect immediately, the scheduling details of the stream definition can be edited which resets the next scheduled run date/time so SaaS batch scheduler can re-determine next scheduled run date/time per new schedule.

# Tips For Setting Up Scheduler Batch Stream Definitions

**Configuring scheduler batch stream definitions to avoid open-ended stream issue**

The scheduler batch stream definitions should always be configured such that the stream "ends" only once all the steps in the stream are completed either by configuring it on the "end of stream" step condition or by defining the inter-dependencies between the steps. This ensures the scheduler batch stream run execution does not stop in the middle of the stream.

**Transferring batch job stream definitions to SaaS batch scheduler**

Customers that are upgrading from prior cloud releases using Oracle Scheduler are recommended to plan transfer and migration of existing batch job streams definitions to SaaS Batch Scheduler. The recommended steps are:

- (Optional Step) Migrate existing batch job stream definitions from production environment to any pre-prod environment.
- In any pre-prod environment, use the transfer option to move the existing batch job stream definition to scheduler batch stream definition (new object for SaaS Batch Scheduler).
- Perform thorough testing of newly created scheduler batch stream definitions.
- Before migrating to the target environments, the batch job stream definitions in the target environment should be canceled(if needed) and deleted.


Copyright © 2026, Oracle and/or its affiliates.

- Migrate the newly created scheduler batch stream definitions by using Content Migration Assistant (CMA) to production. Migration Plans for scheduler batch stream definitions are available for review.

**Jobs that run past midnight must have same business date as other jobs in a stream**

In situations where a scheduler batch stream runs too long and has some jobs run past midnight, all jobs in the scheduler batch stream can be set to run with same business date. Set the business date option value on the scheduler batch stream definition to 'SYSDATE' and the scheduler engine processes all the steps in the scheduler batch stream with same date when the scheduler batch stream started.

**How to split a big stream into multiple smaller stream**

If the Scheduler Batch Stream Definition is too long with many jobs to run, it can be split into multiple smaller streams such that one stream runs after the other. This can be done using a new batch control K1-RNSBS (Run Scheduler Batch Stream). Create a Batch Scheduler Batch Control for every sub stream, specify the batch control as K1-RNSBS, batch parameter name as the sub stream name. Use this Batch Scheduler Batch Control as a last step on the previous stream before it ends so that as soon as the previous stream's steps are completed, the last step will initiate the new sub stream. For example, if you have a stream with jobs A,B,C,B,E,F,G,H. Split the stream into stream 1 with A,B,C,D and stream 2 with E,F,G,H. Create a Batch Scheduler Batch Control with batch control K1-RNSBS, batch parameter name as stream 2. On stream 1 after step D, define a new step that invokes the created Batch Scheduler Batch Control. Note that there are no dependencies between sub-streams. That is each sub stream would be kicked off by the new batch control to run on it's own and the stream that called the sub stream does not wait for it to finish. So such configuration is to be used with only those steps that run in parallel.

> **Note**
>
> Sub-streams follow the main stream's schedule. Schedule on a sub-stream is generally not supported.

**How to configure scheduler batch stream definitions to run only on OUAF working calendar dates**

To setup Scheduler Batch Stream Definition to run only on working days, use the batch control K1-CWCEL. Create a Batch Scheduler Batch Control with this batch control, batch parameter Work Calendar set to required work calendar, Verify that **Run On Business Day** is set to true. Use this Batch Scheduler Batch Control as the first step to run on the stream. Define the stream to end if this step fails. Define the stream's subsequent step to execute if this step succeeds. When the stream runs on a scheduled day, the first step checks if the day falls on a working day as per the work calendar. If the day is not a working day then the stream ends execution ensuring stream runs only on desired dates.

**Scheduler batch stream definitions that check existence of a file**

In situations where a scheduler batch stream definition needs to check if a file has been uploaded/created by another process before it can continue with rest of it's processing, a special batch program K1-OSCFE is available. A stream step can be introduced referring to this new batch program with parameter specifying which object storage bucket to check the file in and also status to return if the file does not exist. This stream step can be added right before the step that has dependency on checking the existence of the file and continue processing or not.


Copyright © 2026, Oracle and/or its affiliates.

#### Handling scheduler batch stream definitions inter-dependencies

In situations where a scheduler batch stream should not run when another scheduler batch stream is running, the inter-dependencies between such scheduler batch stream definitions can be configured by using a special batch program K1-CKCCR. Following steps are needed to setup such configuration.

- The new batch program accepts two parameters script name and script data. Write a CM script that checks whether a dependent job stream is running. If it's running, then the script should terminate with an error.
- On the scheduler batch stream that needs to be paused/hold if another job stream is running, add a new stream step as the first step of the job stream. Use the new batch program on the first step(by setting the CM script as the input parameter) or create a batch scheduler batch control if the batch control is to be used across many job streams with different CM service scripts as input parameters. Set the dependent job stream name as the script data parameter.
- Update the "End Of Stream" step condition such that if first step has failed, end the stream. This ensures the scheduler batch stream does not run if the other dependent scheduler batch stream is already running.
- Update the step(s) intended to execute after the first step, the step condition that if first step succeeded only then these steps should continue.

> **Note**
>
> The special batch program is not restricted to only these situations. It can be used in many custom scenarios by writing the corresponding CM script that terminates if a desired situation is met.

## System Override Date Support

This section provides information on system override date support.

A System Date can be overridden at either system level or user level or both. SaaS Batch Scheduler supports the system date override at system level only. The scheduler batch streams are scheduled and processed including submitting batch jobs on the stream steps per system date override. Any online operations performed by user on the scheduler batch stream operations portal such as skip, pause, cancel etc are also processed per system date override.

> **Note**
>
> When system date is overridden, scheduler batch stream definitions that are already scheduled and have next scheduled date set on them do not automatically get the new scheduled date per the overridden system date. This means the next run would be per the current date/time. The overridden system date reflects on the subsequent runs after the scheduler batch stream runs once. If the system override date needs to take effect immediately, the stream's schedule can be updated. This reset/blanks out the next scheduled run so it gets determined immediately per system overridden date.

# Batch Scheduler Batch Control


Copyright © 2026, Oracle and/or its affiliates.

This section provides information on using batch scheduler batch controls.

- Configuring a New Batch Scheduler Batch Control

## Configuring a New Batch Scheduler Batch Control

In order to configure a new Batch Scheduler Batch Control, from the **Admin** menu, select **Batch Operations**, and select **Batch Scheduler Batch Control**. Use the **Batch Scheduler Batch Control** portal to search for and view existing batch scheduler batch controls.

A Batch Scheduler Batch Control is an entity that holds details of a batch control. Typically a scheduler batch stream definition is defined with batch control specified directly on it. In this case the batch parameters on the batch control are used in batch job submissions. A Batch Scheduler Batch Control can be created when batch parameters need to be customized different from the default batch parameters. The Batch Scheduler Batch Control can then be used on the Scheduler Batch Stream Definition. Only active batch scheduler batch controls can be used on the Scheduler Batch Stream Definition.

# Scheduler Batch Stream Operations

The **Scheduler Batch Stream Operations** portal provides an overview of scheduler batch streams that are currently running or suspended or completed within the last week.

To navigate to Scheduler Batch Streams Operations portal, from the **Admin** menu, select **Batch Operations**, and select **Scheduler Batch Stream Operations**.

By default the Running Search By option is displayed where streams that are running can be searched. Broadcast any of the stream run for more information on status of each of its steps.

Two more Search By options are available to search the scheduler batch streams â€” 1. Suspended 2. Completed

**Running** Search By Option: This option displays Scheduler Batch Stream runs that are In Progress, Stopping, Pausing or Canceling statuses. Refer to Monitoring A Running Scheduler Batch Stream for more information.

**Suspended** Search By Option: This option displays Scheduler Batch Stream runs that are Failed, Stopped or Paused statuses. Refer to Monitoring A Suspended Scheduler Batch Stream for more information.

**Completed** Search By Option: This option displays Scheduler Batch Stream runs that are Ended or Canceled statuses. Refer to Monitoring A Completed Scheduler Batch Stream for more information

- Monitoring a Running Scheduler Batch Stream
- Monitoring a Suspended Scheduler Batch Stream
- Viewing a Completed Scheduler Batch Stream
- Handling Exceptions

## Monitoring a Running Scheduler Batch Stream

A Scheduler Batch Stream is considered Running if it is one of the following statuses.

- In Progress


Copyright © 2026, Oracle and/or its affiliates.

- Stopping
- Pausing
- Canceling
- Ending

You can monitor the progress of a running Scheduler Batch Stream by broadcasting it from the Running search by option search results. The broadcasted zone **View Stream Steps** shows progress of the stream such as what steps have already completed and their status, what is the currently running step(s) along with steps that have not yet started. You can monitor the progress of the stream by refreshing the page or by configuring the page to refresh automatically. To configure automatic page refresh, refer to the **User Portal Preferences** for this portal.

The **View Streams Steps** zone also has information on when the step started and completed, the duration the step ran for and the option to view the batch job run. Clicking on the **View** button displays the batch run details of the batch job.

> **Note**
>
> A stream is set to Ending status if stream is In Progress and the End Of Stream step has ended. This means some of the steps are still either in In Progress or Submitted status. An Ending stream is set to Ended status once no more steps are In Progress or Submitted status.

### Stream Operations

When performing operations on a scheduler batch stream, the operation is not immediately processed real-time. User can only make a request for a **desired** operation. SaaS Batch Scheduler processes these requests and updates scheduler batch stream run accordingly.

Following operations are allowed on the Running scheduler batch streams.

- *Cancel:* A scheduler batch stream run can be canceled while in In Progress or Pausing state. When this operation is performed, scheduler batch stream is set to Canceling status. SaaS Batch Scheduler processes these requests and submits a cancellation request to cancel the underlying batch job. Once the underlying batch job is canceled, the scheduler batch stream is set to Canceled status. A Canceled scheduler batch stream is removed from this Running streams list and is displayed on the Completed streams list.
- *Stop:* A scheduler batch stream run can be stopped while in In Progress state. When this operation is performed, scheduler batch stream is set to Stopping status. SaaS Batch Scheduler processes these requests and submits a cancellation request to cancel the underlying batch job. Once the underlying batch job is canceled, the scheduler batch stream is set to Stopped status. A Stopped scheduler batch stream is removed from this Running streams list and is displayed on the Suspended streams list


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> Note the difference between canceling a scheduler batch stream and stopping a scheduler batch stream. Canceling a scheduler batch stream cancels the steps in progress and cancels the scheduler batch stream so rest of the stream steps do not run permanently. Stopping a scheduler batch stream stops/cancels the stream steps that are In Progress and stops the stream execution temporarily. A stopped stream can be re-started from the Suspended Streams list on the Scheduler Batch Streams Operations Suspended Search By Option. Re-starting a stopped stream re-starts the steps that were stopped and submits the batch jobs associated with those steps to continue rest of the processing of the scheduler batch stream.

- *Pause:* A scheduler batch stream can be paused while In Progress state. The scheduler batch stream is set to Pausing status. SaaS Batch Scheduler processes these requests and pauses the scheduler batch stream once stream steps that are In Progress have completed. While the steps are In Progress, the scheduler batch stream is set to Pausing status. Once steps are complete, the scheduler batch stream is set to Paused status. If the stream steps that are In Progress are the last set of steps to be processed, then there are no steps remaining to pause the scheduler batch stream. Scheduler batch stream is not Paused and completes processing.

> **Note**
>
> Note the difference between stopping a scheduler batch stream and pausing a scheduler batch stream. Stopping a scheduler batch stream cancels the steps that are in progress and stops the stream execution temporarily. Pausing a scheduler batch stream, the steps In Progress continue execution and complete and stream is paused after that. Rest of the stream execution is on hold temporarily.

- *Interrupt:* An interrupt operation is used when the scheduler batch stream is stuck in In Progress state for long period of time and is not progressing. When used, the interrupt operation cancels the steps that are currently In Progress and re-submits the batch jobs on those steps so that the rest of the steps in the scheduler batch stream can be processed.

**Stream Step Operations:**

- *Skip:* A stream step(s) can be skipped by using the skip multi-action button on this zone. Only steps that are in "Not Submitted" state can be skipped.

The stream steps can be in one of the following states:

- Not Submitted
- Submitted
- In Progress
- Ended
- Skipped

# Monitoring a Suspended Scheduler Batch Stream


Copyright © 2026, Oracle and/or its affiliates.

A Scheduler Batch Stream is considered suspended if it's stopped running temporarily. A Scheduler Batch Stream is considered Suspended if it is one of the following statuses.

- Failed
- Stopped
- Paused

You can monitor the progress of a suspended Scheduler Batch Stream by broadcasting it from the Suspended search by option search results. The broadcasted zone **View Stream Steps** shows progress of the stream such as what steps have already completed and their status, what is the currently running step(s) along with steps that have not yet started. You can monitor the progress of the stream by refreshing the page or by configuring the page to refresh automatically. To configure automatic page refresh, refer to the **User Portal Preferences** for this portal.

The **View Stream Steps** zone also has information on when the step started and completed, the duration the step ran for and the option to view the batch job run. Clicking on the **View** button displays the batch run details of the batch job.

### Stream Operations

When performing operations on a scheduler batch stream, the operation is not immediately processed real-time. User can only make a request for a **desired** operation. SaaS Batch Scheduler processes these requests and updates scheduler batch stream run accordingly.

Following operations are allowed on the Running scheduler batch streams.

- *Resume:* A scheduler batch stream run can be resumed from Paused state. When this operation is performed, scheduler batch stream is set to In Progress status. SaaS Batch Scheduler processes these requests and starts processing the rest of the steps in the scheduler batch stream. A resumed scheduler batch stream is removed from this Suspended streams list and is displayed on the Running streams list.
- *Restart Stream:* A scheduler batch stream run can be restarted while in Failed or Stopped state. When this operation is performed, scheduler batch stream is set to In Progress status. SaaS Batch Scheduler processes these requests and starts processing the rest of the steps starting with Failed or Stopped steps. A restarted scheduler batch stream is removed from this Suspended streams list and is displayed on the Running streams list
- *Cancel:* A scheduler batch stream run can be canceled while in any one of the Suspended states. When this operation is performed, scheduler batch stream is set to Canceling status. SaaS Batch Scheduler processes these requests and in case of Failed scheduler batch stream, submits a cancellation request to cancel the underlying batch job. Once the underlying batch job is canceled, the scheduler batch stream is set to Canceled status. A Canceled scheduler batch stream is removed from this Running streams list and is displayed on the Completed streams list.

**Stream Step Operations:**

- Skip: A stream step(s) can be skipped by using the skip multi-action button on this zone. Only steps that are in "Not Submitted" state can be skipped.

The stream steps can be in one of the following states:

- Not Submitted
- Submitted


Copyright © 2026, Oracle and/or its affiliates.

- Ended
- Failed
- Stopped
- Skipped

# Viewing a Completed Scheduler Batch Stream

A Scheduler Batch Stream is considered Completed if it is one of the following statuses.

- Ended
- Canceled

You can view completed Scheduler Batch Stream by broadcasting it from the Completed search by option search results. The broadcasted zone **View Stream Steps** shows status of the stream such as what steps have already completed and their status, what is the currently running step(s) along with steps that have not yet started. You can monitor the progress of the stream by refreshing the page or by configuring the page to refresh automatically. To configure automatic page refresh, refer to the **User Portal Preferences** for this portal.

The **View Stream Steps** zone also has information on when the step started and completed, the duration the step ran for and the option to view the batch job run. Clicking on the **View** button displays the batch run details of the batch job.

The stream steps can be in one of the following states:

- Not Submitted
- Submitted
- Ended
- Failed
- Skipped
- Canceled

# Handling Exceptions

### Errors in Scheduler Batch Stream

A batch job submitted as part of a Scheduler Batch Stream can run into error with one of its threads being in error. It could be because of business scenario or environment issue etc. In such a case, the corresponding Scheduler Batch Stream stops execution and its step ends in failed state. The cause for the failure is noted on the Additional Details column on the stream step. You can fix the issue and restart the stream via the multi-select **Restart Stream** option on


Copyright © 2026, Oracle and/or its affiliates.

the scheduler batch stream operations completed query option. Restarting a stream will result in the Scheduler Batch Stream continuing execution from the step that failed and finish through to end.

> **Note**
>
> Once a Scheduler Batch Stream fails, its future scheduled runs no longer execute until the failed run is addressed.

**Scheduler Batch Stream stuck in running status**

A Scheduler Batch Stream can be stuck in Running status as one of its steps is in running status for too long. In this situation there are various options.

*Cancelling a scheduler batch stream*: The corresponding batch job submission is stuck in pending status as the thread pool name set on the batch control is invalid. In such case, the running Scheduler Batch Stream can be cancelled via the multi-select **Cancel** option. The thread pool name can then be set to a valid value and the Scheduler Batch Stream can be submitted manually via the **Run Manually** option on the Scheduler Batch Stream Definition or wait until it's scheduled to run next.

*Interrupting a scheduler batch stream*: The corresponding batch job submission is stuck in started status while the respective batch run or batch threads have ended. In this case, the running scheduler batch stream can be interrupted via the multi-select **Interrupt** option. This sets the scheduler batch stream and respective steps in the "Failed" state. The **Restart Stream** multi-select option can then be used to re-start the stream to continue execution from the step that failed and complete processing of stream.

# Batch Queues Overview

Batch Queues is a new portal where users can view the status and progress of various batch jobs in the system. The portal provides visibility for users into what batch jobs are queued or running on existing thread pools DEFAULT and NOCACHE.

Users can view each of the batch job's details and broadcast them to view list of threads, their status and records details.

The list of batch jobs can be filtered by status, thread pool and batch control to view desired batch jobs.

# Batch Alerts

This chapter provides information and requirements on batch alerts in support of Oracle Utilities Cloud implementations.

- Batch Alerts Overview
- Alerting On A Batch Job Stream
- Alerting On A Batch Job


Copyright © 2026, Oracle and/or its affiliates.

# Batch Alerts Overview

Batch Alerts provide alerting and monitoring mechanisms for batch and batch job streams. Alerts are raised when desired batch jobs or batch job streams do not perform as expected.

Currently there are few level of service algorithms on batch controls to indicate how the corresponding batch jobs are doing.

New level of service algorithms are now introduced on the batch controls.

In addition, level of service algorithms are introduced on batch job streams to determine overall health of batch job streams.

New installation options health check algorithms are introduced for health component batch and batch job stream. Each health component's health check algorithm will invoke their level of service algorithms and return the overall results

Overall results from health check algorithms are displayed on the health check portal for monitoring purposes to view overall health of the system.

To receive alerts on overall health of the system, use the alert probe - bash script external utility that probes the system and sends email notifications to configurable set of email addresses.

The alert probe can be run on windows or linux machines. Alerts can be customized for errors and warnings. A paging alert is sent in case of errors. A non-paging alert is sent in case of warnings.

The frequency at which alerts are raised is configurable.

> **Note**
>
> The alert probe is provided on an use-as-is basis, no support is provided for it.

# Alerting On A Batch Job Stream

To receive alerts on an active batch job stream, edit the batch job stream and add the desired level of service algorithm on it.

> **Note**
>
> Only algorithm types are delivered for Batch job stream level of service algorithms. A corresponding algorithm must be created before it can be added onto the batch job stream

Following level of services are currently supported on batch job streams

- Batch Job Stream has not started within 'X' minutes


Copyright © 2026, Oracle and/or its affiliates.

- Batch Job Stream has failed
- Batch Job Stream ran too long (relative run)

## Alerting On A Batch Job

Additional level of services are included on batch job that will be part of alerting mechanism

- Batch Job has processed low number of records
- Batch Job throughput is too low
- Batch Job ran too long (relative run)
- Batch Job has not started within 'X' minutes

# ILM Dashboard for Cloud Implementations

This chapter provides information on ILM Dashboard in support of Oracle Utilities Cloud implementations.


- ILM Dashboard Overview
- ILM Dashboard - Main Tab
- ILM Dashboard - Partitions Tab
- ILM Dashboard - Drop A Partition


## ILM Dashboard Overview

ILM Dashboard is a feature that provides an all-in-one portal to view Information Lifecycle Management (ILM) configurations and manage ILM tasks in one place. ILM Dashboard has two tabs - Main tab and Partitions tab.

Main tab displays information on current ILM configuration, if any, current retention period of maintenance objects and status of ILM batch processes

Partitions tab displays information on current partition data in the system for all maintenance objects including the sub-partition data.

## ILM Dashboard - Main Tab

The ILM Dashboard - Main tab has information on current ILM configurations, retention period on maintenance objects and ILM batch processes current status.

**ILM Configuration:** Displays the current ILM configuration in the system as well as the default retention period for all ILM eligible maintenance objects.

**Retention Period Overridden Maintenance Objects:** Displays the list of ILM eligible maintenance objects where the retention period is overridden and their current retention period.



Copyright © 2026, Oracle and/or its affiliates.

**ILM Batch Jobs Status:** Displays the list of ILM Batch Controls and their latest batch job/run details. ILM Batch Controls that never ran even once are displayed first followed by the ones that ran in descending order of their last run date/time.

**ILM Batch Runs:** Displays the batch runs for the ILM batch control that was broadcasted on the "ILM Batch Jobs Status" zone. The batch runs are displayed sorted by end date/time with recently ended ones displayed first. Each run's first three batch parameter values are displayed on it's run. Customers can navigate to respective batch job submission page to view list of all parameters and their values.

# ILM Dashboard - Partitions Tab

The ILM Dashboard - Partitions tab displays information on current partitions in the system for all maintenance objects that are ILM enabled.

**Maintenance Objects:** Displays a drop down with list of maintenance objects that are ILM enabled. Selecting a maintenance object displays the maintenance object along with it's partition information such as all partitions Total Size (In GB) and Total Records Based On Last Stats. If no maintenance object is selected, all maintenance objects are displayed along with their partition information.

Checking the PMAX partitions with data displays only those maintenance objects and their PMAX partitions information that have data in the PMAX partitions. If a maintenance object and PMAX partitions with data is selected, only the PMAX partitions information of the PMAX partitions that have data in it are displayed for the selected maintenance object.

**Partitions:** Displays list of partitions and their details for the selected maintenance object. The partitions are sorted by partition name latest month first. By default, the last twelve months and Pmax partition are displayed by default. User can further search for historical partitions using the filter options upto last four years.

For each partition, it's Compression Type is displayed. Compression Type for each displays "Refer Sub-partition" if the partition does not have compression but one of the sub-partitions does.

For a maintenance object, at any given time, the oldest partition can be requested to be dropped via the "Drop Partition" button. A new "Partition Drop Request" object is then created for this request to go through the approval process. Refer to Dropping A Partition for more information on request to drop partitions and it's approval process.

**Sub-partitions:** Displays list of sub-partitions and their details for the selected partition. The sub-partitions are sorted by sub-partition name latest month first.

**Partition Drop Requests:** Displays list of partition drop requests along with it's current status and applicable actions for the current user. Only displayed if there are partition drop request records. Displays partition drop requests for the selected maintenance object, if none is selected then all partition drop requests for all maintenance objects are displayed.

By default, only partition drop requests in Pending and Error status are displayed. For partition drop requests in other statuses, user can use the filter options. Refer to Dropping A Partition for additional information.

# ILM Dashboard - Drop A Partition

The ILM Dashboard - Partitions tab displays information on current partitions in the system for all Maintenance Objects that are ILM enabled.


Copyright © 2026, Oracle and/or its affiliates.

For a Maintenance Object, at any given time, the oldest partition of the primary table can be requested to be dropped via the "Drop Partition" button by a **Requester** user. A new "Partition Drop Request" object is then created for this request in Pending status and a To Do is created for an Approver user so the request goes through the approval process.

**Requesters**

A user must be setup as a Requester user in order to drop a primary table's oldest partition for a selected Maintenance Object at any given point. Please refer to Administrative User Guide for more information on how to setup these users.

> **Note**
>
> A pre-requisite exists for dropping a primary table's oldest partitions with data. The crawler Batch Control associated with the Maintenance Object must be run first prior to requesting the partition drop, so that relevant data is appropriately marked for the partition to be dropped.

> **Note**
>
> When oldest partition of the primary table for a maintenance object is requested to dropped, the oldest partitions of child tables for that maintenance object are also dropped. This is based on the referenced partitions setup for the maintenance object. If the maintenance object does not have referenced partitions setup, then only the primary table's oldest partition is dropped.

The "Partition Drop Request" object is initially in Pending status and displayed to **Approver** users in the "Partition Drop Requests" zone.

**Approvers**

A user must be setup as an Approver user for them to approve any Partition Drop Requests.Please refer to Administrative User Guide for more information on how to setup these users.

**Viewing Drop Partition Requests**

Only the Requester or Approver can view Drop Partition Requests based on their access. By default, only Pending or Error Drop Partition Requests are displayed. User can use the filter critera option to display all Drop Partition Requests or search for ones with different statuses. Depending on the user's access, the available and applicable actions are displayed to the user.

When the Approver views list of Partition Drop Requests, they have the option of either approving or rejecting the request.

Approving a request results in Partition Drop Request in In Progress status and completing the To Do previously created for that Approver. A batch job submission for Batch Control K1-ILMDR (ILM Automation - Drop Partition) is also initiated to drop the requested partition. Once the partition is successfully dropped, the Partition Drop Request transitions to Completed final status. Otherwise, if there are any errors, the Partition Drop Request transitions to Error status.

Rejecting a request requires specifying a reason as to why the request was rejected. Possible reasons for rejecting a request are 1. Approver determines that partition that is requested to drop is required and so it should not be dropped. 2. Incorrect request to drop a partition was made where Requester accidentally requested wrong partition to be dropped. Once the Approver rejects the request, the Partition Drop Request transitions to Rejected final status.


Copyright © 2026, Oracle and/or its affiliates.

When the Partition Drop Request is in Error status, the Requester has the option to re-try the request again or cancel the request. In the case of re-try option, Requester can review the errors, take any necessary steps needed and and re-try the Partition Drop Request again. For example, if the pre-requisite step was not taken prior to requesting a partition drop, the data in the partition will not be marked for dropping. In this case, an error would be generated when the partition with that data is requested to be dropped. Requester can review the error to confirm the reason and run the crawler Batch Control associated with the Maintenance Object to mark partition data to be dropped. Once it is complete, Requester can re-try the Partition Drop Request again.

A Partition Drop Request in Pending status can be canceled by the Requester user if desired. A Requester user can also cancel the Partition Drop Request that is in Error status if they wish to not re-try the request.

**Recommendations**

It is recommended that dropping partitions is tested thoroughly in non-production environments first before performing the same operation in a production environment.

**Tips**

- Once a partition is dropped, the data is also removed from the database and there is no way to restore the data back into the environment. It is imperative that customers review the partition and the respective data before requesting and approving the Partition Drop Request.
- Customers can optionally choose to raise an SR(Service Request) to request a database copy prior to dropping a partition

# Oracle Cloud Infrastructure Identity and Access Management Integration

This chapter provides information about integration with Oracle Cloud Infrastructure Identity and Access Management for Oracle Utilities cloud services.

> **Note**
>
> This section does not apply to all implementations. Contact your system administrator to confirm if your implementation includes Oracle Utilities Cloud Service Foundation and Oracle Cloud Infrastructure Identity and Access Management (IAM) integration.


- Identity and Access Management Integration Overview
- Configuring Identity and Access Management Integration


## Identity and Access Management Integration Overview

Oracle Cloud Infrastructure Identity and Access Management (IAM) with Identity Domains provides identity and access management functionality for the Oracle Utilities cloud services and supports single sign on (SSO) and identity federation capabilities.

It protects both online and webservices API access to the Oracle Utilities cloud service. Refer to the Detailed Oracle Cloud Services Administration guide for the instructions on how to



Copyright © 2026, Oracle and/or its affiliates.

configure and manage online and API access and maintain users, groups, application roles and integration OAuth clients.

In IAM, an identity domain is provided with each service subscription. Once provisioned, the identity domain is administered exclusively by the client.

Security administrator uses identity domains to manage application users, who are given access to a one or multiple application environment(s). The security administrator may also assign another user to an administrative role in the identity domain and delegate user management privileges.

Newly created users receive an account activation email from IAM, and must reset their password.

The integration with IAM supports Just-In-Time User Provisioning.

When a user attempts to access the utility application URL, the user is redirected to an identity domain for authentication. Once the user is successfully authenticated, they are redirected back to the application and the authentication token is evaluated and validated. The system then checks if the user already exists in the application, and if not, it triggers the Just-In-Time User Provisioning logic that creates and activates the new user.

The **F1-IDMUser** business object is used for creation of new user records sourced from the external identity management integrations.

For the integration with Oracle Cloud Infrastructure Identity and Access Management, the algorithm is defined on the **F1-IDMUser** business object to determine the Template User whose information should be copied to the newly created user.

- Just In Time User Provisioning

# Just In Time User Provisioning

Just-In-Time User Provisioning is a process of creating a new user record upon first successful login. The implementation may follow either one of the main approaches:

- **Just-In-Time with Coarse-Grained User Authorization** allows security administrator to immediately begin working on the authorization and access setup, while other ("business") users might wait until the proper access and authorization is configured for them in the application.
- **Just-In-Time with Fine-Grained User Authorization** provides both security administrator and the business user with instant access to the appropriate application functionality.
- Coarse-Grained User Authorization (Default)
- Fine-Grained User Authorization

## Coarse-Grained User Authorization (Default)

With course-grained user authorization, Just-In-Time Provisioning creates users with two authorization levels:

- **Security and Access Administrator.**
  - **Users with an administrative role in the identity domain** are provisioned using a **Security Administrator Template User**. The user gains the access to all transactions and services related to application security and user authorization setup
- **Authenticated User**


Copyright © 2026, Oracle and/or its affiliates.

- **Users with no administrative role(s) in the identity domain** are provisioned using a **Minimum Access Template User**. The user gains access to *My Preferences* page only. The security and access administrator completes the authorization and access setup manually.

No additional configuration effort is needed if your implementation chose to follow this approach. The scope of the actual security access for both Security Administrator and Authenticated User is customizable. Refer to Override Default Access for more information.

## Fine-Grained User Authorization

With fine-grained user authorization,Just-In-Time Provisioning creates users with multiple authorization levels, according to their Identity Domain's Group membership and based on utility application configuration:

- **Security and Access Administrator**
  - **Users with an administrative role in the identity domain** are provisioned using a **Security Administrator Template User**. The user gains the access to all transactions and services related to application security and user authorization setup
- **Business/Administrative Users** with access to a specific business and/or administrative functions in the application
  - **Users that are members of a Group in the identity domain** are provisioned using a **Template User mapped to this Group**. The user gains access to the application according to the Template User's setup.
- **Authenticated User**
  - **Users with no administrative role(s) in identity domain** and **no membership in any of the identity domain's Groups** are provisioned using the **Minimum Access Template User**. The user gains access to *My Preferences* page only. The security and access administrator completes the authorization and access setup manually.

The mapping between identity domain's Groups and Template Users is stored in theIdentity and Access Management Integration Configuration.

# Configuring Identity and Access Management Integration

The integration configuration in the utility application has to be done in conjunction with the corresponding setup in the Identity and Access Management system.

- Configuring Just In Time User Provisioning
- Customizing the Integration

## Configuring Just In Time User Provisioning

In order to configure user authorization for Just in Time Provisioning, perform the following steps:

- Create User Groups that represent most typical business and/or administrative roles in the application
- Create Template Users that represent typical access level for major business and/or administrative functions


Copyright © 2026, Oracle and/or its affiliates.

- Request your customer's identity domain Administrator to create Groups in IAM with Identity Domains that will correspond to the Template Users create above
- Setup the mapping between the identity domain's Groups and Template Users using Identity and Access Management Integration Configuration
- Template Users and User Groups
- Identity and Access Management Integration Configuration

## Template Users and User Groups

- Define User Group(s) that represent the major business and/or administrative responsibilities in the application
- Define Template User(s) that represent the typical level of access to the application functionality

## Identity and Access Management Integration Configuration

The Identity and Access Management Integration Configuration master configuration stores the definitions related to the integration with Oracle Cloud Infrastructure Identity and Access Management.

For Just-In-Time User Provisioning, configure the **User Provision Rules**:

- Map the Groups defined in the identity domain to the Template Users defined in the application.

# Customizing the Integration

The product is delivered with out-of-the-box support for Just in Time User Provisioning with Coarse-Grained Authorization. Possible customization scopes are:

- Customize the default access level for Coarse-Grained Authorization. See Override Default Access Level for details.
- Configure the Just in Time User Provisioning with Fine-Grained Authorization.
- Override Default Access Level
- Override Logic to Determine Template User

## Override Default Access Level

Both Security Administrator and Minimum Access Template Users are specified as parameters for the K1-IDCSUSRTM algorithm .

To override the default access level:

- Create new Template Users and setup user groups for default administrative access
- Create new Template Users and setup user groups for default minimum access

Create a new version of the K1-IDCSUSRTM algorithm and specify the Template Users created above as the algorithm's parameters


Copyright © 2026, Oracle and/or its affiliates.

## Override Logic to Determine Template User

The algorithm that determines the Template User is implemented as a pre-processing algorithm that determines the Template User on the **F1-IDMUser.** business object.

To override the logic used to determine the template user:

- Create a custom algorithm type and algorithm and use custom logic to determine the Template User for provisioning.
- Deactivate the existing product algorithm and specify the custom algorithm instead. Note that the algorithm that determines the Template User should be plugged before the algorithm that populates the data from the Template User.

# Oracle Cloud Object Storage

Oracle Cloud Object Storage (Object Storage) is a part of Oracle Cloud Infrastructure Storage Services.

Oracle Cloud Object Storage can be accessed and managed using Oracle Cloud Infrastructure Console or programmatically using Command Line Interface (CLI), REST APIs or Java. The Object Storage that is used by the system is owned and managed by customers.

Oracle Utilities Cloud Services use Object Storage as the file storage location for all file related integrations. Batch processes that need to read or write files as input will access Object Storage.

The system requires some configuration in order to allow it to connect to the Object Storage and have access to the files that need to be processed by it. The following chapter describes the required configuration in order to connect to Object Storage as well as provides information on how to reference files in Object Storage in batch processes.

Please refer to Oracle Cloud Infrastructure documentation for more information on Object Storage.

- File Storage Configuration for Object Storage
- Referencing Files on Object Storage

## File Storage Configuration for Object Storage

In order for the system to connect to Object Storage a File Storage Extendable Lookup has to be created with the following information:

- The Oracle Cloud Infrastructure tenancy details, which can be obtained from the Oracle Cloud Infrastructure Console(available to customers who purchased Object Storage).
- The API Key that is used by the system to connect to that Object Storage.

Refer to product documentation on "File Storage" for more details.


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> - Each File Storage Location configuration refers to a combination of Object Storage Tenancy, Compartment and User ID.Since customers can choose to create many Compartments in their Object Storage, access to each such Compartment will require a separate File Storage configuration.
> - It is also recommended that each File Location configuration will reference a **different** Cloud Infrastructure User ID.This can limit, for example, the system's access to production files from a non-production environment.
> - API Keys are managed using Key Rings. Key Rings can be shared between File Storage Extendable Lookups but the Keys in these Key Rings cannot be shared across system environments. For example, keys generated for the Development environment **cannot** be used for the Testing or Production environments.Refer to the following topics in the Framework Administrative User Guide:
>
>   **External File Storage**, **Key Rings**.

# Referencing Files on Object Storage

Access to files is typically required in Batch processes. Some batch processes include direct reference to the files they need to process (for input or output) while some processes rely on additional configuration (for example, the Content Migration Assistant (CMA) Import and Export processes that rely on the CMA Master Configuration Settings).

### Direct Reference

Reference to Object Storage can be used anywhere that a file location is allowed. The format is:

file-storage://<File Location>/<Bucket>/<Filename.ext>

- <File-Location>: The File Storage Extendable Lookup value that is defined for that file.It should include all the details the system needs to access the Object Storage location for that file.
- <Bucket>: The equivalent of a Folder in the Oracle Cloud Object Storage.Since the folder structure in Object Storage is flat, Buckets cannot include other Buckets and therefore only one Bucket can be noted here.

### CMA Master Configuration

CMA Master Configuration includes the location of files for export and import processing:

- Import Folder: should reference a file location for an Object Storage Bucket(i.e. just the file location without the a file name, for example: "file-storage://CMALocation/CMA-Files"), "CMALocation" is the File Location Extendable Lookup name.
- Export Folder: should reference a file location for an Object Storage Bucket (uch as "file-storage://CMALocation/CMA-Files").


Copyright © 2026, Oracle and/or its affiliates.

> **Note**
>
> In order to take advantage of the automation of migration of configuration data from one system environment to another via the Process Automation Tool, the Import and Export directories have to point to the same location!

**Conversion Data Upload Processes**

The batch process that loads customer data files into the cloud (K1-CNVLD) using SQL Loader has two parameters that describes where to get the input data files and where to store the output files:

- inputFileStorage: should reference a file location for an Object Storage Bucket (such as "file-storage://CONVLocation/Input"), "CONVLocation" is the File Location Extendable Lookup name.
- outputFileStorage: should reference a file location for an Object Storage Bucket (such as file-storage://CONVLocation/Output).

# Maintaining Customer Code

The system supports a process to import custom Java code to be deployed in the cloud under certain use cases.

> **Note**
>
> Refer to the Cloud Services Implementation Guide for information about how to use this functionality.

The Customer Code portal allows you to review the status of the lifecycle of the process to import code and allows you to review errors that may have been received.

To view a customer code record, navigate using **Admin**, then **Implementation Tools**, then **Customer Code**. You are brought to a query portal with options for searching for customer code. Once a record has been selected, you are brought to the maintenance portal to view and maintain the selected record.


Copyright © 2026, Oracle and/or its affiliates.